

Flight Operations Team (FOT) Standard Operating Procedures (SOPs) For The ECS Project

White Paper

September 1997 (FINAL)

SUBMITTED BY

Rick Hudson
EOSDIS Core System (ECS) FOT Manager

CONCURRENCE

Bob Kozon
NASA EOS Flight Operations Director (FOD)

Hughes Applied Information Systems
Upper Marlboro, Maryland

This page intentionally left blank.

Abstract

The Standard Operating Procedures (SOPs) provide the ECS Flight Operations Team (FOT) with a detailed set of instructions for the management and utilization of the configured items (CIs) associated with FOS. This document also provides IST users with information unique to IST operations that is not covered in the FOS Operations Tools Manual. This is a living document which contains or will contain procedures defining the protocol for ground operations including data analysis, anomaly reporting and resolution, TDRS scheduling, and configuration management. The individual SOPs contained within this document describe the manner in which such operational functions will be executed on a daily basis. For additional details on the specific tools and FOS features to facilitate the development of individual CIs, reference the FOS Operations Tools Manual.

Keywords: User's, FOT, IOT, Manual, tools, IST, FOS, Operations, EOC

This page intentionally left blank.

Contents

Abstract

Contents

I. Introduction

- 1. Configuration Management (CM)**
- 2. Project Database (PDB)**
- 3. Constraints**
- 4. Activity Definition**
- 5. Baseline Activity Profile (BAP)**
- 6. ECL Command Procedure**
- 7. Display Builder**
- 8. Relative Time Command Sequences (RTCS)**
- 9. Derived Parameters**
- 10. Algorithms**
- 11. Inhibit Identifiers Management**
- 12. Telemetry Monitor (TMON)**
- 13. Flight Software (FSW)**
- 14. Decision Support System (DSS)**
- 15. FOS Configuration Files**

16. Rooms

I. Introduction

I1 Purpose

The purpose of the Flight Operations Team (FOT) Standard Operating Procedures (SOPs) for the ECS Project is to provide the FOT and IST users with information on the SOP protocol.

This document also provides IST users with information that is unique to IST operations.

The FOT SOPs for the ECS Project serves as the users' guide for developing Configured Items (CIs) to be delivered to the EOS Operations Center (EOC). The FOT SOPs is a living document which will be appended to as new SOPs are identified. The most recent version (new or modified) of each SOP will be distributed via the WWW or the FOS anonymous login.

I2 Organization

This paper is organized such that each SOP starts a new section. Each section includes:

- a. Originator
- b. Purpose
- c. Background
- d. Responsibilities
- e. Prerequisites/Constraints
- f. Procedures
- g. References
- h. Appendix

I3 Review and Approval

This White Paper is an informal document approved at the FOT Office Manager level. It does not require formal Government review or approval; however, it is submitted with the intent that review and comments will be forthcoming.

Since the information contained in this document addresses the specific policies and procedures by which the FOT will accomplish flight and ground ops-related functions, the FOT Manager and the NASA Flight Operations Director (FOD) shall be the only signatories to the document. However, given that the specifics of the operational policies described herein will affect various other elements involved in EOS mission operations, individual SOPs will be disseminated externally with each review cycle. In some cases, the information provided in an individual SOP may augment information contained within an Instrument Operations Interface Control Document (OIICD) or flight systems documentation. Therefore, the Instrument Operations Teams (IOT) and the spacecraft vendor shall be responsible for reviewing those SOPs which are deemed relevant to their interests. Sign-off by the NASA FOD will not take place until major

issues with the document (or with individual SOPs within the document) raised by external elements are negotiated and resolved.

Questions regarding technical information contained within this Paper should be addressed to the following ECS contact:

Ronald Jones, FOT Ground System Manager <rjones@eoc.ecs.nasa.gov>

Questions concerning distribution or control of this document should be addressed to:

Bob Kozon, NASA Flight Operations Director (FOD) <bob.kozon@gsfc.nasa.gov>
Goddard Space Flight Center
Code 505
Greenbelt, Maryland 20771

1. FOT Configuration Management SOP

11 Originator

1.1.1 Name: R. Jones Date: 3/14/97

1.1.2 Revised by: R. Jones Date: 8/25/97

12 Purpose

The purpose of the Configuration Management (CM) Standard Operation Procedure (SOP) is to document the Flight Operations Team (FOT) Operations CM Policy, CM Plan, and CM Processes, as well as the CM Implementation Procedures for AM-1 Mission Unique Operations Configured Items (CIs). Mission Unique CIs are essentially procedures, mission data and operation agreements. If there are any discrepancies between this document and the Configuration Management Plan for the ECS Project (102-CD-002-001), the latter takes precedence.

Mission Independent CIs will not be covered in this SOP. Mission Independent CIs are multi-mission CIs which are applicable to all ECS Missions. They include ECS hardware, software and firmware components. CM for Mission Independent CIs are documented in the Configuration Management Plan for the ECS Project (102-CD-002-001).

13 Background

1.3.1 Configuration Control Boards Hierarchy

Critical to the implementation of mission operations is the process used to ensure documentation control, system configuration control, change approval, and status reporting and tracking. The configuration management process backs-up the operations philosophy for the program. The CM policies and process are governed and enforced by a hierarchy of Configuration Control Boards (CCBs):

1. Mission To Planet Earth (MTPE) CCB
2. Flight Project CCB
3. Spacecraft Integrator CCB
4. ESDIS Project CCB
5. FOT Operations CCB

The relationship of these CCBs is shown in figure 1.3.1-1.

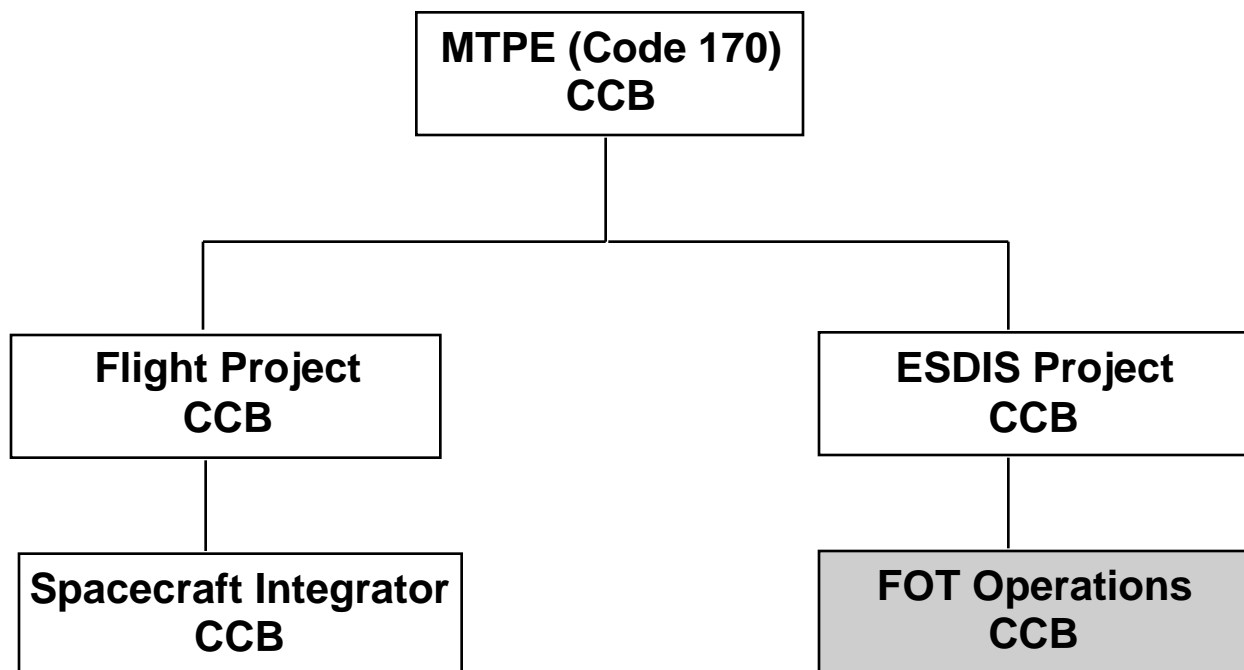


Figure 1.3.1-1: CCB Hierarchy

1.3.2 MTPE CCB

The MTPE Project CCB is responsible for reviewing and evaluating all changes to proposals which affect the system performance, impact cost or delay delivery dates.

1.3.3 Flight Project CCB

The Flight Project CCB has been established to provide configuration control of the spacecraft related documents, software, and hardware. The CCB is responsible for reviewing evaluations of each proposed change from all aspects that:

1. Impacts any spacecraft support documents: Requirements documents, ICDs, Plans, Schedules, Pre-Launch Test Plans or Post Launch Test Plans.
2. Impacts any spacecraft software and/or hardware that have been placed under AM-1 Project configuration control.
3. Impacts any Mission Operations Normal Command Procedures, Contingency Procedures, Activities or Mission Operations Database.

1.3.4 Spacecraft Integrator CCB

The Spacecraft Integrator/Lockheed Martin Missiles & Space (LMMS) CCB is the governing entity of all AM-1 development activities at Valley Forge, PA. This CCB is subordinate to the Flight Project CCB. Configuration baselines are established and managed throughout the program's life cycle.

1.3.5 ESDIS Project CCB

The ESDIS Project CCB has been established to provide configuration control of the mission operations-related documents, ground software, and hardware. The CCB is responsible for

reviewing evaluations of each proposed change from all aspects (e.g., technical, interface, operational, logistics) that:

1. Impacts any Mission Operations support documents: Requirements documents, ICDs, Plans, Schedules, Operations guidelines, Operations Reports, Failure Diagnostic Logic Diagrams or Post Launch Test Plans.
2. Impacts any software and/or hardware that have been placed under AM-1 Project configuration control and affect AM-1 spacecraft subsystems.
3. Impacts any Mission Operations Normal Command Procedures, Contingency Procedures, Activities or Mission Operations Database.

1.3.6 FOT Operations CCB

The FOT Operations CCB has been established to provide configuration control of FOS Operational Data, FOT Operational Procedures and FOT Operational Agreements. This CCB is subordinate to the ESDIS Project CCB. The CCB is responsible for reviewing and evaluating proposed changes to CIs. CIs are a collection of Data/Procedures/Agreements used to support operations at the EOC and managed by the FOT DBA.

1.3.7 FOT Operations CM Policy

The success of the AM-1 mission depends not only on the successful development of the AM-1 Bus and Instruments, but also on the error-free Mission Operations and readiness of the Mission Operations Ground System and Mission Operations Crews. The FOT Operations CM Process plays the most critical role to ensure that, during any phase of the AM-1 Mission Operations, only fully authenticated Command Procedures and associated data and information are being used by the FOT and Instrument Operations Teams (IOTs). The FOT Operations CM Policy includes the following:

1. AM-1 Mission Operations shall use only formally configured AM-1 Mission Operations Command Procedures and associated systems, data, and information in conducting operations during all phases of the AM-1 Mission.
2. This Policy governs all the AM-1 Mission Unique Operations Configuration Items.
3. This Policy establishes the AM-1 FOT Operations CCB and grants it the authority of configuration control of all the AM-1 Mission Operations Configured Items (CIs).
4. This Policy requires that any changes to the baselined AM-1 Mission Operations CIs shall be formally initiated and processed via a Configuration Change Request (CCR), and the changes shall be documented, verified, validated, reviewed, and approved by the FOT Operations CCB via established FOT Operations CM Processes and Procedures.

14 Responsibilities

The FOT Operations CCB will convene and approve all AM-1 Mission Unique Operations CIs. It is the FOT Operations CCB's responsibility to:

1. Control the baseline configuration of all AM1 Mission Unique Operations CIs.
2. Review all CCRs to ensure the integrity of all the baseline configurations.
3. Approve CCRs for the implementation of configuration changes to the controlled CIs.

The FOT Operations CCB members are as follows:

- I. Chairperson: FOT Ground System Manager
- II. Board Members:
 - A. FOT Flight Operations Manager

- B. FOT Spacecraft Manager
- C. NASA Flight Operations Director (FOD)

The FOT Database Administrator (DBA)/Configuration Manager (CM) will manage and implement CCB approved CCRs.

15 ~~Prerequisites~~ Constraints

Before a CI can be reviewed by the CCB:

1. A CCR needs to be written.
2. The CI needs to be tested.
3. CI Listing needs to be provided with the Validation Form as an attachment.

16 Procedures

1.6.1 Operations CI List

The following Data CIs can be created from the EOS Operations Center (EOC) or an Instrument Support Terminal (IST), and are under FOT configuration control:

1. Project Database (PDB): Commands, Telemetry, Activities, Constraints
2. Baseline Activity Profiles (BAPs)
3. ECL (ECS Control Language) Command Procedures (CPs)
4. Displays: Pages/Rooms
5. Relative Time Command Sequence (RTCS)
6. Derived Parameters (pseudo A mnemonics - simple equations)
7. Algorithms (pseudo B mnemonics - complex equations)
8. Inhibit Identifiers (commands, RTCSs, Telemetry Monitors [TMONs])

The following Data CIs can only be created at the EOC:

- I. TMON
- II. Flight Software Code & Tables:
 - A. Spacecraft Control Computer (SCC)
 - B. Command and Telemetry Interface Unit (CTIU)
 - C. Solid State Star Tracker (SSST)
- III. Decision Support System (Expert System States & Rules)
- IV. Flight Operations Segment (FOS) configuration definitions (User ID, passwords, constants, event messages, etc.)

The following CIs are FOT created procedures and agreements involving other elements.

1. SOPs: **SOPs will be written for all CIs listed above.** These SOPs will cover the creation, update, and maintenance of all CIs. They will also address aspects of CM which are specific to the individual CI.
2. Operations Agreements (OA): IOTs, Flight Dynamics Division (FDD), Network Control Center (NCC), EOS Data and Operations System (EDOS), etc.

1.6.2 Data Configured Items (CI) Description

Data CI's are those data files used by operations to configure/control the ground system and spacecraft. They are configuration controlled by FOT. CI's are built at the EOC User Stations or IST Workstations by FOT/IOT personnel using FOS provided tools. CI's are broken into the following categories:

1. Spacecraft Bus
2. Instruments
3. Ground

The FOT will generate the spacecraft and ground configured items. The IOTs will create the instrument CIs with assistance/support from the FOT as required.

1.6.3 Data Configured Items (CI) Test Checkout

CIs must be validated and tested before submitting to the CCB (Figure 1.6.4-1). Following are the steps to check out a CI:

- I. User defines CI in their own personal account (IST Workstation / EOC User Station)
- II. User Syntax, Validates / Constraint Checks CI
- III. User tests CI by themselves or in conjunction with FOT
 - A. Against FOS
 1. example 1: Page Display
 2. example 2: Create Activity, Schedule Activity, Display on Timeline, Generate DAS & CMS Loads
 - B. Against SSIM (i.e.: Command Procedure)
- IV. User Submits CI to FOT Operations CCB for review
- V. Approved CIs migrated to EOC Ops Area by the FOT DBA
 - A. Select group of CIs used by the FOT/IOT to fly the mission.
 - B. PDB Tools use to move data from the CM Tables to the EOC Operations area.
 - C. UNIX commands used to copy UNIX Files from the FOT DBA CM Directory to appropriate EOC Operations directory.

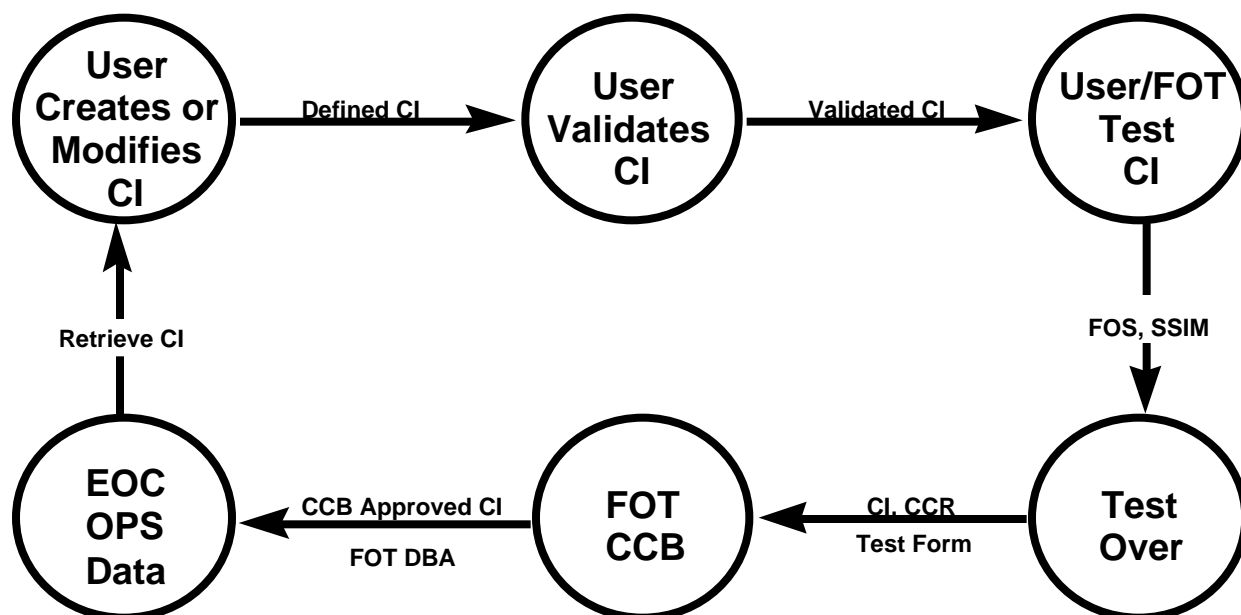


Figure 1.6.4-1: CM Flow Diagram

1.6.4 FOT Operations CCB Process:

The FOT Operations CCB is the governing entity of all the AM-1 Mission Unique Operations CIs.

- I. Author must submit the following items:
 - A. CI Listing (Report)
 - B. FOT AM-1 CCR, see Appendix 1.8.1
 - C. FOT AM-1 Validation Form, see Appendix 1.8.2
- II. Formal approval of a new or modified CI requires the following four signatures:
 - A. Author
 - B. FOT Operations Manager: Demonstrates that the FOT Operations Lead Engineer has reviewed the CI and that it is consistent with the Ground System and the Operations Plan
 - C. FOT Spacecraft Manager: Demonstrates that the CI is consistent with the current technical subsystem understanding and it's interaction with other spacecraft subsystems.
 - D. NASA FOD: Demonstrates that NASA has reviewed the CI, and concurs that its functional implementation is consistent with subsystem engineering design.
 - E. CCB Chairperson: Accepts or rejects CCR based on discussion during the CCB meeting. Responsible for implementation of accepted CCRs and resolution of rejected CCRs.

1.6.5 CCR/Validation Form Processing

The following are the detailed steps of the CM Process for processing the CCR and Validation Form for any CI:

- I. The FOT/IOT Originator initiates the CCR (Appendix 1.8.1) to develop or make changes to a CI.
 - A. The Originator completes the following CCR page information:
 1. Requested by: Can be same as Originator or different (i.e. Originator's Manager/Supervisor)
 2. CI Type: CI from CI List (See 1.6.2)
 3. CI ID: CI's Name
 4. Subsystem: GNC, CDH, AST, CEA, MOD, MOP, MIS, Ground, etc.
 5. Revision and Document Traceability: If this is a revision of a CI that has been CCB-approved (i.e., it has the FOD signature and "ACCEPTED" in the Action Field), then the Originator must fill in "A" in the Revision Field and must write in BASELINE CCR # (original CCR number) in the Document Traceability Field.
 6. Reason for Change: Any further revisions must have the appropriate REV (e.g. B, C, etc.) in the Revision Field and the traceability back to the Baseline CCR # in the Document Traceability Field (e.g. Baseline CCR #, REV A CCR #, etc.)
 7. Comments: Optional
 - B. The CCB Chairperson will complete:
 1. CCR Number and Date.
 2. Circle Accept or Reject CCR as well as Initial and Date CCR.
- II. The Originator prints out the CCR page, initials and dates the Originator Field. This page is kept as the sign-off page, and all of the original signatures must be on this ONE page.

- III. The appropriate FOT/IOT member then develops the CI.
- IV. The FOT Spacecraft Manager and FOT Operations Manager conduct an internal review of the CI. This involves a telecon with a relevant IOT as required. The FOT/IOT revises the CI to address red-lines made during the internal review and resubmits it to the FOT Spacecraft Manager and FOT Operations Manager for signatures.
- V. If there are no major revisions, the FOT/IOT creates the CI in a designated Working Area on FOS/IST and Validates the CI using: FOS/IST, Spacecraft Simulator (SSIM), Spacecraft (S/C).
- VI. The validator (does not have to be the same as the CCR Originator) completes the Validation Form providing all requested information (Appendix 1.8.2).
 - A. CI Type, CI ID, Revision, Subsystem, CCR Number, CCR Date: Same as (1) above
 - B. Data Base and FOS/IST Versions: Indicate Versions used when testing the CI.
 - C. Checking: Yes/No
 - D. Means of validation: Yes/No
 - E. Procedures Called and Arguments Checked: Indicate any values or passed parameters used in the validation.
 - F. Location of Validation Documentation: Locations such as PC/Workstation Directory/Filename, File Cabinet, etc.
 - G. Comments: Optional
- VII. The Originator then creates the configuration package which contains the CCR sign-off page, the original CI with comments from the FOT Managers, the final CI with the corrected red-lines high-lighted and initialed by the Originator and FOT Managers, and the Validation Page.
- VIII. FOT Operations CCB reviews the CCR Package. The FOD will sign for NASA. It is the FOD's responsibility to consult with the appropriate IOT member or Spacecraft Integrator before signing the CCR. If the CCB approves the CCR then the CI will be moved from the FOS/IST Working Area to the FOS Operations Area. If the CCB rejects the CCR, it is returned to the Originator.
- IX. At this point the CM Process for the CCR is complete and the new or updated CI is under strict configuration control. Any further modifications to the CI will require a new CCR to be initiated.

17 References

- 1. Data Format Control Document for the Earth Observing System (EOS) Flight Operations Segment (FOS) Project Data Base, Volumes I and II (505-10-35-1 and 505-10-35-2)
- 2. Configuration Management Plan for the ECS Project (102-CD-002-001)
- 3. Flight Operations Segment (FOS) Operations Tools Manual for the ECS Project (609-CD-005-003)
- 4. Operations Scenario Document for the ECS Flight Operations Segment (605-CD-003-001)
- 5. EOS AM-1 Mission Operations Review
- 6. FOS Critical Design Review

18 Appendix

- 1. FOT AM-1 CCR
- 2. FOT AM-1 Validation Form

181 FOTAM-1 CONFIGURATION CHANGE REQUEST

Requested by:

CCR Number:
CCR Date:

CI Type:
CI ID:
Revision:
Subsystem:

Document Traceability:

Reason for Change:

	Initials	Date	Action
CCR Originator:	_____	_____	
FOT Ops Man.:	_____	_____	
FOT S/C Man.:	_____	_____	
NASA FOD:	_____	_____	
CCB Chair:	_____	_____	Accept / Reject (circle one)

Comments:

182 FOTAM-1 VALIDATION FORM

Validated by:

Date Validated:

CI Type:

CI ID:

Revision:

Subsystem:

CCR Number:

CCR Date:

Data Base Version:

FOS/IST Version:

Syntax Checked?

Nominal Path Checked?

All Logical Paths Checked?

Run against FOS/IST On:

Run against SSIM On:

Run against S/C On:

Procedures Called:

Arguments Checked:

Location of Validation Documentation:

Comments:

2. Project Database (PDB) SOP

21 Originator

2.1.1 Name: R. Jones Date: 4/1/97

2.1.2 Revised by: S. Haugh Date: 8/4/97

22 Purpose

The purpose of the Project Database (PDB) Standard Operating Procedure (SOP) is to document the Flight Operations Team (FOT) PDB Plan and Processes, as well as the PDB Implementation Procedures. If there are any discrepancies between this SOP and the Data Format Control Document (DFCD) for the Earth Observing System (EOS) Flight Operations Segment (FOS) Project Data Base Volumes I and II (505-10-35-1 and 505-10-35-2), the latter takes precedence.

The PDB contains telemetry, command, activity and constraint definitions necessary to support mission planning, spacecraft commanding and telemetry processing. **Activity and constraint definition details will not be covered in this SOP.** They will be documented in the **Activity SOP** and **Constraint SOP**, respectively.

23 Background

The PDB consists of definition files from the Lockheed Martin Corporation's (LMC), AM-1 Spacecraft Manufacture, Integration and Test (I&T) database and from the FOT. Initially, LMC will provide the telemetry and command definitions for the spacecraft and instrument subsystems. This information is derived from the I&T database. The FOT, using database edit tools provided at the EOS Operations Center (EOC), will provide activity and constraint definitions and any additional information needed to support telemetry and command processing. The method of transfer of the definition files from LMC will be via the Internet. Each new release of information from the I&T database will be sent to a dedicated directory at the EOC. Additionally, the FOT Database Administrator (DBA) will be notified via E-mail of the transfer.

PDB files provided by the FOT will be defined at the EOC. Constraint definitions will be generated using the Constraint Definer Tool (CDT) and activity definitions will be generated using the Activity Definer Tool (ADT).

The Instrument Operations Teams (IOTs) will be provided with the capability to update the FOS PDB for their instrument-specific definitions using database edit tools. This function will be provided as part of the Instrument Support Toolkit (IST). The IST will serve as the primary mechanism for performing command and telemetry updates to the FOS PDB IOT definitions at the point when the Valley Forge (VF) I&T database becomes obsolete. Before the FOS PDB is frozen (VF I&T database still exists), **IOT modifications to the FOS PDB** using the IST database edit tools **will be flagged**. These modifications will not be overwritten with the VF I&T database values. Operations and I&T values are not always the same. The IOT changes to the FOS PDB are made because the values are the ones needed for operations and therefore will be kept. The IOTs will create all activity and constraint definitions using their ISTs.

The FOS PDB processing and how it relates to the VF I&T database as well as the FOT and IOTs is shown in Figure 2.3-1.

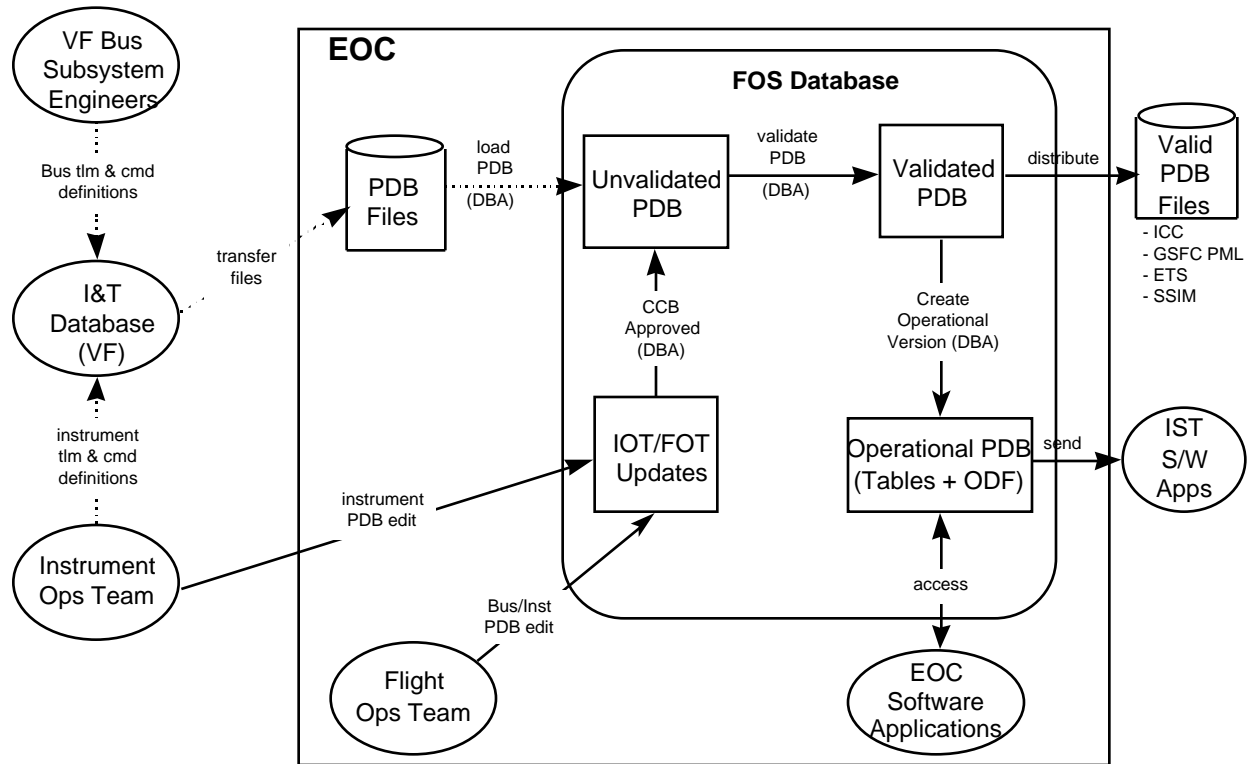


Figure 2.3-1: PDB Processing

Key: Before PDB Frozen
 _____ After PDB Frozen (~3 months before launch)

24 Responsibilities

2.4.1 Spacecraft Contractor / FOT / IOT

The EOS AM-1 PDB defines the telemetry, commands, activities and constraint definitions. These definitions are provided by the spacecraft contractor, the FOT and the IOT. The information provided in the DFCD governs the definition files received by and processed at the EOC. Table 2.4.1-1 lists each file in the PDB.

Table 2.4.1-1. List of PDB Files (1 of 2)

Filename	Description
TELEMETRY	
t1m_packet_XXX.pdb	Telemetry Packet Specification PDB
t1m_parm_XXX.pdb	Telemetry Parameter Specification PDB
t1m_desc_XXX.pdb	Telemetry Description PDB
t1m_calcurve_XXX.pdb	Telemetry EU Conversion Specification PDB
t1m_polyconv_XXX.pdb	Polynomial Coefficients Specification PDB
t1m_lineconv_XXX.pdb	Linear Coefficients Specification PDB
t1m_rylim_XXX.pdb	Red/Yellow Limit Specification PDB
t1m_limsel_XXX.pdb	Limit Selection Specification PDB
t1m_delta_XXX.pdb	Delta Limit Specification PDB
t1m_dstate_XXX.pdb	Discrete State Specification PDB
t1m_context_XXX.pdb	Context Dependent Specification PDB
t1m_derived_XXX.pdb	Derived Telemetry Specification PDB
t1m_constant_XXX.pdb	Telemetry Constant Specification PDB
t1m_lgdesc_XXX.pdb	Telemetry Long Description PDB
COMMANDS	
cmd_parm_XXX.pdb	Command Parameter Specification PDB
cmd_desc_XXX.pdb	Command Description PDB
cmd_fixdata_XXX.pdb	Command Fixed Data Word Specification PDB
cmd_vardata_XXX.pdb	Command Variable Data Word Specification PDB
cmd_verify_XXX.pdb	Command Execution Verification (CEV) PDB
cmd_pstate_XXX.pdb	Prerequisite State Specification PDB
cmd_lgdesc_XXX.pdb	Command Long Description PDB
cmd_tbldef_XXX.pdb	Table Definition PDB
cmd flddef_XXX.pdb	Table Field Definition PDB
cmd_mmask_XXX.pdb	Memory Masking Definition PDB
cmd_maskgrp_XXX.pdb	Memory Masking Group Definition PDB
ACTIVITIES	
act_def_XXX.pdb	Activity Definition PDB
act_dirdef_XXX.pdb	Activity Directive Definition PDB
act_ecldir_XXX.pdb	Activity Complex ECL Directive Definition PDB
act_eclcmd_XXX.pdb	Activity ECL Command Procedure Definition PDB
act_eclcom_XXX.pdb	Activity ECL Comment Directive Definition PDB
act_cmdddef_XXX.pdb	Activity Command Definition PDB
act_cmdsuflld_XXX.pdb	Activity Command Subfield Definition PDB
act_dfltsuflld_XXX.pdb	Activity Default Command Subfield Definition PDB
act_intsuflld_XXX.pdb	Activity Integer Subfield Definition PDB
act_flsuflld_XXX.pdb	Activity Float Subfield Definition PDB

Table 2.4.1-1. List of PDB Files (2 of 2)

act_dscsubfld_xxx.pdb	Activity Discrete Subfield Definition PDB
act_dscsubvv_xxx.pdb	Activity Discrete Subfield Valid Values Definition PDB
act_funcsub_xxx.pdb	Activity Function Subfield Definition PDB
act_cmdproc_xxx.pdb	Activity Command Procedure Definition PDB
act_cmplxact_xxx.pdb	Complex Activity Child Reference PDB
act_modetran_xxx.pdb	Activity Mode Transition Definition PDB
act_mode_xxx.pdb	Activity Mode Definition PDB
act_resdef_xxx.pdb	Activity Resource Definition PDB
act_rescmd_xxx.pdb	Activity Resource Command Definition PDB
act_respwr_xxx.pdb	Activity Resource Power Definition PDB
act_resbuff_xxx.pdb	Activity Resource Buffer Definition PDB
act_reshier_xxx.pdb	Activity Resource Hierarchy Definition PDB
act_schedhier_xxx.pdb	Activity Scheduling Resource Hierarchy Definition PDB
act_schedres_xxx.pdb	Activity Scheduling Resource Definition PDB
CONSTRAINTS	
con_bitrule_xxx.pdb	Bit Rule Definition PDB
con_comrule_xxx.pdb	Comment Rule Definition PDB
con_noexistrule_xxx.pdb	No Exist Rule Definition PDB
con_offsetrule_xxx.pdb	Offset Rule Definition PDB
con_prerule_xxx.pdb	Prerule Definition PDB
con_postrule_xxx.pdb	Postrule Definition PDB
con_satisfier_xxx.pdb	Satisfier Constraint Definition PDB
con_scalarrule_xxx.pdb	Scalar Rule Definition PDB
con_subfld_xxx.pdb	Subfield Constraint Definition PDB
con_subfldbit_xxx.pdb	Subfield Bit Constraint Definition PDB
con_opmode_xxx.pdb	Operational Mode Specification PDB
con_opmodetran_xxx.pdb	Operational Mode Transition Specification PDB
con_event_xxx.pdb	Event Specification PDB
con_activity_xxx.pdb	Activity Constraint Specification PDB
con_consume_xxx.pdb	Consumable Constraint Specification PDB

Note: xxx represent the version number (000-999) associated with the PDB.

2.4.2 FOT Database Administrator (DBA)

The FOT DBA will manage and implement Configuration Control Board (CCB) approved PDB Configuration Change Requests (CCRs).

The FOT DBA:

1. Maintains error log
2. Performs backup and recovery
3. Incorporates PDB modifications
4. Performs PDB ingest, validation, and operational data generation

25 Prerequisites/Constraints

The following sections discuss the PDB prerequisites and constraints.

2.5.1 PDB Updates

One prerequisite is that updated PDB files from the spacecraft contractor have been received or PDB updates by the FOT/IOT have been approved by the CCB.

2.5.2 Web Page

Another prerequisite is that Netscape has been successfully configured and the DB Utilities web page is available.

2.5.3 PDB Processing Time

PDB processing time will vary depending on which parts of the PDB are processed. The PDB generation process allows the activities and constraints parts of the PDB to be processed independently of the command and telemetry parts of the PDB, as explained in Sections 2.6.5 and 2.6.6. It takes approximately 9 hours to process a **complete** PDB release including commands, telemetry, activities, and constraints. The breakdown by part is the following:

- I. Approximately 1 hour to process the commands part of the PDB.
- II. Approximately 7 hours to process the telemetry part of the PDB.
- III. Approximately 30 minutes to process the activities part of the PDB.
- IV. Approximately 30 minutes to process the constraints part of the PDB.

26 Procedures

2.6.1 PDB Transfer

The PDB generation process is performed in a non-real-time, interactive environment. The PDB generation process begins with the transfer of PDB files derived from the latest version of the I&T database. The PDB files containing **telemetry and command definitions** will be sent to a dedicated area within the FOS system. These definition files will be loaded into the telemetry and command PDB directory within the FOS system by selecting "PDB Load" on the Database Utilities (DBA) window. Each new release of the PDB files will replace the existing release.

2.6.2 Constraint Definitions

Constraint definitions are provided by the FOT/IOT through the use of the CDT. Reference the **Constraint SOP**.

2.6.3 Activity Definitions

Activity definitions are provided by the FOT/IOT through the use of the ADT. Activity definitions will be appended to the activity PDB as needed. Reference the **Activity SOP**.

2.6.4 PDB Edit

The database edit function will allow addition, deletion and modification of the contents of the definitions files once loaded into the PDB directory at the EOC. Information not provided by LMC through their I&T database but required for EOC operations can be added by the FOT using this function. Additionally, definition files can be updated and invalid data can be deleted. The IOT can provide changes to the PDB through the use of a database edit tool. Once updates have been made, the IOT must notify the DBA through E-mail. The DBA will be responsible for loading these changes into the PDB after being approved by the FOT Operations CCB. Following are the steps (Table 2.6.4-1) the user will employ to modify, add, or delete commands or telemetry definitions in the PDB:

Table 2.6.4-1. PDB Edits

Step	Description	Notes
1	Initiate the Database Utilities (User) window through a web browser (Netscape).	
2	Select "Database Access".	
3	Select "PDB Parameters".	
4	Select "Command Parameters" or "Telemetry Parameters".	
5	Choose the desired PDB Record to edit.	
6	Enter the requested information in the text fields displayed.	
7	Select the Access method (Add, Update, or Remove).	
8	Enter password and submit request.	Modifications to database definitions require a password
9	Access record definitions to verify modification was made. Go to the Command Parameters page or the Telemetry Parameters page and select the desired record.	

2.6.5 PDB Validation

Once corrections and changes have been made to the data definitions, the DBA may invoke validation of the PDB by selecting "PDB Validation" on the Database Utilities (DBA) window. This step will ensure the accuracy of the definitions used for operations. By selecting "View the PDB Validation Output File", the DBA may review the results of the validation. The process of editing and validating the PDB may be repeated until the validated PDB is acceptable for operational use.

- I. Validation includes the following:
 - A. syntax checking
 - B. verification of values
 - C. cross-checking of related definition files
- II. Updates to the constraint and activity PDB may occur independent of a new release of the I&T database. For this reason, the PDB generation process provides validation of the constraint and activity PDB, independent of the other definition files. Three methods for validation of the PDB will exist at the EOC.
 - A. The first method provides for validation of the entire PDB, i.e., validation of the telemetry, command, constraint and activity definitions. This process reflects a specific order which is required to support the integrity of the definitions.

- B. The second method provides for validation of the constraint definitions only.
- C. The third method provides for validation of the activity definitions only.

2.6.6 PDB Generation

Upon acceptance of the validated PDB, the DBA may invoke the operational data generation process by selecting “Operational Data Generation” followed by “Operational Data File (ODF) Generation” on the Database Utilities (DBA) window. These steps will prepare the definitions for operational use by FOS software applications. Operational data generation may occur for the entire validated PDB as in the validation process, for only the validated constraint PDB, or for only the validated activity PDB. During this process, definitions requiring conversion will be reformatted to support the users need. A version number will be generated reflecting the validated PDB being used, i.e., version 2.1.2 would indicate the second release of the I&T database, with the first version of validated constraints associated with that release and the second version of validated activities associated with that release.

2.6.7 PDB Distribution

As part of the operational data generation process, the new version of PDB definitions used for operations will be dumped from the database and distributed. The distribution list for the operational PDB definitions includes the GSFC Program Maintenance Library (PML), EOSDIS Test System Multimode Portable Simulator (ETS MPS), and Spacecraft Simulator (SSIM). Additionally, the validated PDB output will be transferred to ASTER and the ISTs.

1. The DBA will send E-mail informing the ICC and ISTs that a new release of the PDB is available and will become operational on a specific date.
2. The ICC and ISTs will download the PDB (ODFs) by polling a directory at the EOC or from a web page (TBD).

2.6.8 PDB Reporting

Authorized users of the data base may select to view information maintained in the operational PDB. The user may access information by PDB file type or by a unique value such as mnemonic. This information may be viewed on a display or can be provided in hard copy form. Following are the steps (Table 2.6.8-1) the user will use to generate, view, or print command and telemetry PDB reports.

Table 2.6.8-1. PDB Report Generation

Step	Description	Notes
1	Initiate the Database Utilities (User) window through a web browser (Netscape).	
2	Select the “Database Reports” option to generate Command or Telemetry PDB reports.	
3	Select “Generate Reports”.	
4	Select “PDB Reports”.	
5	Select “PDB Command Complete Report” or “PDB Telemetry Complete Report”.	
6	Click “Okay” to generate the Command or Telemetry Report.	
7	Go back to the Database Reports page and select the	

	“View Reports” option to see the generated report.	
8	Select “PDB Command Reports” or “PDB Telemetry Reports”.	
9	Enter the report name in the text field box and click “Okay”.	
10	Go back to the Database Reports page and select the “Print Reports” option to print the generated report.	These are large files so, the user may wish to cancel the job once a fair amount of data is printed
11	Select “PDB Command Reports” or “PDB Telemetry Reports”.	
12	Enter the report name in the text field box and click “Okay”.	
13	Go back to the Database Reports page and select the “Generate Reports” option to generate a partial report.	
14	Select “PDB Reports”.	
15	Select the “PDB Command Partial Report” or “PDB Telemetry Partial Report” to produce a partial Command or Telemetry PDB report by specifying a mnemonic.	
16	Enter the desired Command or Telemetry mnemonic from the Report previously generated and click “Okay”.	
17	Go back to the Database Reports page and select the “View Reports” option to see the generated report.	
18	Select “PDB Command Reports” or “PDB Telemetry Reports”.	
19	Enter the report name in the text field box and click “Okay”.	
20	Go back to the Database Reports page and select the “Print Reports” option to print the generated report.	
21	Select “PDB Command Reports” or “PDB Telemetry Reports”.	
22	Enter the report name in the text field box and click “Okay”.	

27 References

1. Data Format Control Document for the Earth Observing System (EOS) Flight Operations Segment (FOS) Project Data Base, Volumes I and II (505-10-35-1 and 505-10-35-2)
2. Configuration Management Plan for the ECS Project (102-CD-002-001)
3. Flight Operations Segment (FOS) Operations Tools Manual for the ECS Project (609-CD-005-003)
4. Operations Scenario Document for the ECS Flight Operations Segment (605-CD-003-001)
5. EOS AM-1 Mission Operations Review
6. FOS Critical Design Review
7. Activity SOP
8. Constraints SOP

28 Appendix

N/A

3. Constraints SOP

31 Originator

3.1.1 Name: R. Jones Date: 4/1/97

3.1.2 Revised by: R. Jones Date: 8/25/97

32 Purpose

The purpose of this Standard Operation Procedure (SOP) is to document the Flight Operations Team (FOT) Constraint Plan and Processes, as well as the Constraint Implementation Procedures. The Flight Operations Segment (FOS) and Instrument Support Toolkit (IST) Constraints will be defined in the Project Database (PDB).

33 Background

3.3.1 Constraints Overview

Constraints are divided into three categories: Activity-Level, Command-Level and Prerequisite State Checking. Command-Level constraint checking is performed on command procedures and on commands within the activity definitions. Activity-Level constraint checking will be performed during activity scheduling. In addition, each constraint may be classified as either "Hard" or "Soft". A hard constraint must be resolved before the Absolute Time Command (ATC) Load or Ground Script can be generated. A soft constraint will be flagged for operations consideration. During ATC load generation, the FOT will assume Instrument Operations Team (IOT) approval of existing soft constraints. Prerequisite State Checking is performed during real-time supports. Each command can have up to 4 analog or discrete telemetry parameters which must be within their specified range before the command will be transmitted.

3.4 Responsibilities

3.4.1 Source of Constraints

The information for the constraints are provided by the Spacecraft Contractor, the FOT and IOT. One or more of the following will be used to obtain constraints:

- I. AM-1 subsystem engineers
- II. AM-1 I&T engineers
- III. AM-1 I&T procedures
- IV. AM-1 documents
 - A. Spacecraft Flight Systems Manual (bus)
 - B. Flight System Operations Manual
 - C. On-Orbit Operations Manual
 - D. Operations Interface Control Documents (instruments)
- V. FOT/IOT personnel

3.4.2 Constraints Defined in the Project Database (PDB)

The EOS AM-1 PDB contains the constraint definitions. The database will include the definition of whether a Activity-Level or Command-Level constraint is “hard” or “soft”. Prerequisite State Specification records will be defined in the PDB for identifying the command/telemetry mnemonic combination. The definitions are provided by the FOT and IOTs. The FOT is responsible for the spacecraft bus and ground constraints. The IOTs are responsible for their own instrument constraints. The information provided in the Data Format Control Document (DFCD) governs the definition files received by and processed at the EOC. Table 3.4.2-1 describes the constraint files in the PDB.

Table 3.4.2-1. List of Constraint PDB Files

Filename	Description
COMMAND CONSTRAINT RECORDS	
con_comrule_xxx.pdb	Comment Rule Definition
con_noexistrule_xxx.pdb	No Exist Rule Definition PDB
con_scalarrule_xxx.pdb	Scalar Rule Definition PDB
con_bitrule_xxx.pdb	Bit Rule Definition PDB
con_offsetrule_xxx.pdb	Offset Rule Definition PDB
con_prerule_xxx.pdb	Prerule Definition PDB
con_postrule_xxx.pdb	Postrule Definition PDB
con_satisfier_xxx.pdb	Satisfier Constraint Definition PDB
con_subfiled_xxx.pdb	Subfield Constraint Definition PDB
con_subfieldbit_xxx.pdb	Subfield Bit Constraint Definition PDB
ACTIVITY CONSTRAINT RECORDS	
con_opmode_xxx.pdb	Operational Mode Specification PDB
con_opmodetran_xxx.pdb	Operational Mode Transition Specification PDB
con_event_xxx.pdb	Event Specification PDB
con_activity_xxx.pdb	Activity Constraint Specification PDB
con_consume_xxx.pdb	Consumable Constraint Specification PDB
PREREQUISITE STATE CHECKING	
cmd_pstate_xxx.pdb	Prerequisite State Specification PDB

Note: xxx represent the version number (000-999) associated with the PDB.

3.4.3 FOT Database Administrator (DBA)

The FOT DBA will manage and implement CCB approved constraint CCRs.

3.4.4 ASTER Constraints

Since ASTER has its own Instrument Control Center (ICC), internal ASTER constraints should be checked by the ICC. All “hard” constraints should be checked at the FOS as well as the ICC. For “soft” constraints, redundant checking at both the EOC and the ICC is not necessary.

3.4.5 Conflicts

Constraint conflicts between Instruments and/or the Spacecraft Bus Subsystems will be checked by the FOS. If the FOT and IOTs can not resolve the conflict, then the NASA Flight Operations Director (FOD) will be responsible for elevating the constraint issue. The FOD and Project Scientist will undertake to resolve the constraint.

35 Prerequisites/Constraints

3.5.1 Web Page

A Web Navigator (i.e. Netscape) has been successfully configured and the DB Utilities web page is available.

3.5.2 PDB Generation Time

It takes approximately 30 minutes to generate the constraints part of the PDB (Version).

3.5.3 Hard Constraint

Hard constraints impact the health and safety of the Spacecraft Bus and Instruments and will not be allowed in the spacecraft stored command load (ATC load).

3.5.4 Soft Constraint

Soft constraints will not affect the spacecraft health and safety, but in nominal situations, should be avoided. Soft constraints serve as a warning to IOTs, but it is permissible to allow them into the spacecraft stored command load.

36 Procedures

3.6.1 Create Command-Level and Prerequisite State Checking Constraints

Command-Level and Prerequisite State Checking Constraints should be defined for each command mnemonic in the Project Database.

The FOT and IOTs will define Command-Level Constraints using the Constraint Definer Web Page. These constraints need to be defined before creating ECL Procedures or Activities.

The FOT and IOTs will define Prerequisite State Checking using the Command Prerequisite State Web Page. Prerequisite State Checking provides the transmission criteria for commands and is performed during real-time supports.

3.6.2 Create Procedure or Activity

The FOT or IOT will create a Procedure (using Procedure Builder) or Activity (using Activity Builder) using the FOS/IST tools. For more detail see the ECL Procedure and Activity SOPs.

- I. Syntax Check Procedure or Activity.
- II. Validate Procedure or Activity. At this point, the Procedure or Activity is passed to the FOS Command Management System (CMS) where Command-Level Constraint Checking is performed.

- A. If “Hard” then the constraint must be resolved. This could include a PDB change.
- B. If “Soft” then the definer of the Activity or Procedure must decide whether they want to override the constraint or continue (Yes or No?).
- C. When a Procedure or Activity is determined to be command constraint free it will be considered validated.

3.6.3 Create Activity-Level Constraint(s)

The FOT and IOT will define Activity-Level Constraints using the Constraint Definer Tool. First the FOT and IOTs must develop activities. The next step is to create Activity-Level Constraints for those activities.

3.6.4 Schedule Activity

Activity-Level constraints checking (automatic) will be performed when the FOT or IOT schedule an activity. The Activity can be schedule on the “What-If” or “Master” schedule. Activities can be scheduled using: time, a Baseline Activity Profile (BAP), or the event pool. The FOS Planning and Scheduling software will evaluate the planned activity (specifically, its relationship to other scheduled activities, orbital events, and modes) to determine if any constraints violations exist.

- I. If “Hard” then the constraint must be resolved. This could include a PDB change.
- II. If “Soft” then the FOT/IOT scheduler must decide whether they want to override the constraint or continue (Yes or No?).
- III. Constraints between Instruments and the Spacecraft Bus Subsystems will be checked by the FOS.

3.6.5 Create Detailed Activity Schedule (DAS)

The FOT creates the DAS from the FOS Planning and Scheduling (PAS) Window and passes the Activity/Baseline Activity Profile (BAP) List to FOS CMS. The CMS receives the Activity/BAP List and expands it into a Command List.

Command-Level Constraint Checking is now done on all the commands for that particular operational day. During ATC load generation and ground script generation, CMS software will evaluate the command rules to determine if the ATC load or ground script should be generated.

- I. If “Hard” then the constraint must be resolved. A “Hard” constraint will cause the ATC Load Generation and Ground Script Generation to fail. The FOT or possibly the IOT must resolve by going back to the PAS Scheduling Window.
- II. If “Soft” then the FOT scheduler must decide whether they want to override the constraint or continue (Yes or No?). FOT will assume IOT approval of existing soft constraints.
 - A. If No, then the FOT must resolve by going back to the PAS Scheduling Window.
 - B. If Yes, then the ATC Load Generation and Ground Script Generation will proceed.

3.6.6 Absolute Time Commands (ATC)

The ATC load will be uplinked to the Spacecraft daily. The ATC processor (in spacecraft software) uses the absolute time tag associated with each command in its list of absolute time commands to determine when the command should be distributed to the appropriate instrument or spacecraft subsystem.

3.6.7 Ground Script

The Ground Script is executed by the ECS FOT and contains routine, planned and real-time activities that need to be performed in the EOC. The ground script contains real-time spacecraft and instrument commands, spacecraft and instrument loads, command verification, and configuration/re-configuration of the EOC ground system.

When Prerequisite State Checking is enabled the FOS software evaluates real-time telemetry parameters to determine if the real-time command should be issued.

37 References

1. Data Format Control Document for the Earth Observing System (EOS) Flight Operations Segment (FOS) Project Data Base, Volumes I and II (505-10-35-1 and 505-10-35-2)
2. Configuration Management Plan for the ECS Project (102-CD-002-001)
3. Flight Operations Segment (FOS) Operations Tools Manual for the ECS Project (609-CD-005-003)
4. EOS AM-1 Mission Operations Review
5. FOS Critical Design Review
6. Activity SOP
7. Command Procedure SOP

38 Appendix

N/A

4. Activity Definition SOP

4.1 Originator

4.1.1 Name: Ed Rodriguez Date: 4/28/97

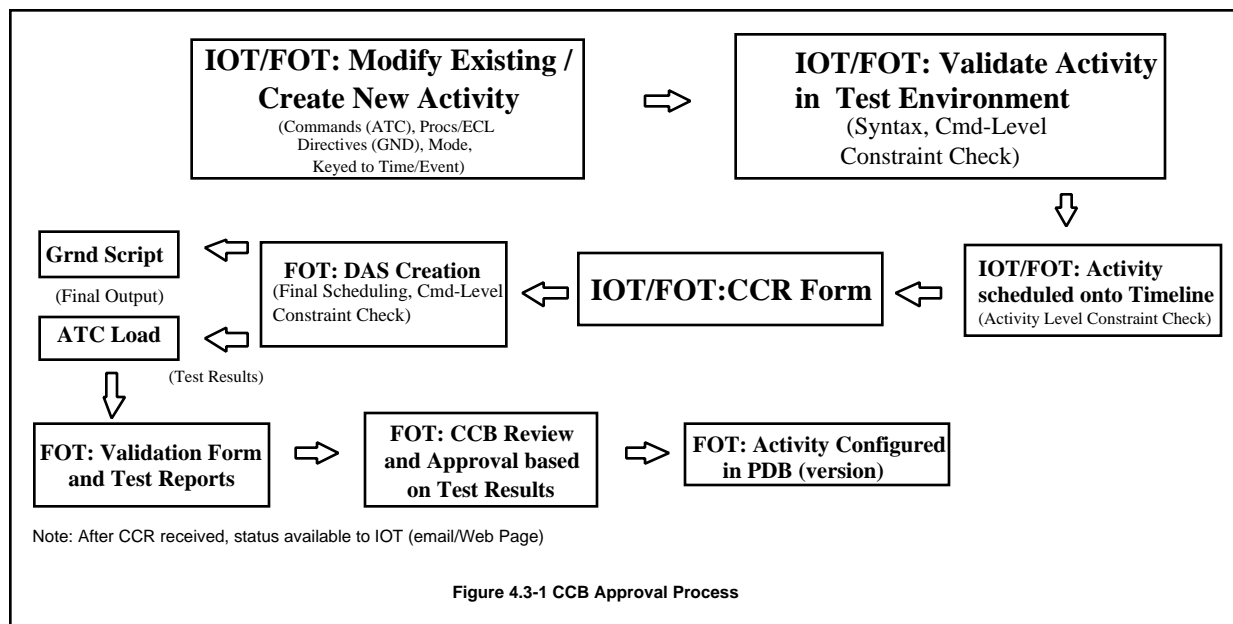
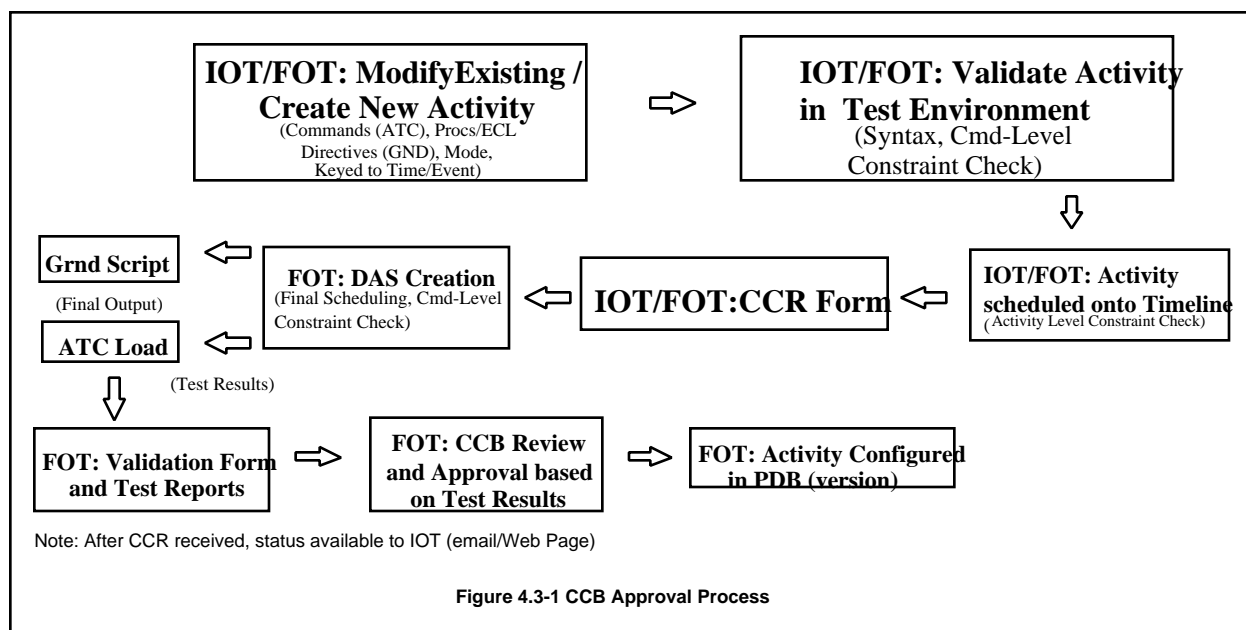
4.1.2 Revised by: Ed Rodriguez Date: 7/28/97

4.2 Purpose

The purpose of this Standard Operating Procedure (SOP) is to document the procedure for creating and/or modifying Activity Definitions to support both instrument and spacecraft operations. This SOP provides the user with a description of what Activity Definitions are, and how they are created, approved and incorporated into the Project Data Base (PDB). Although the details of the FOT Configuration Management (CM) process are covered in a CM-specific SOP, this Activity Definition SOP will discuss aspects which are Activity-specific.

4.3 Background

The FOS Planning and Scheduling subsystem (PAS) provides users with an Activity Definer tool to create Activities needed to support instrument and spacecraft on-orbit operations. Both IOT's and FOT will have the capability to generate, maintain, and schedule Activities for execution onboard the spacecraft. IOT's will have access to the PAS Activity Definer via Instrument Support Toolkits (IST's) located at their facility. Activities are comprised of individual Commands (ATC and Ground), ECL Directives and Command Procedures which will be used by both the IOT's and the FOT to command Instrument and Spacecraft subsystems, respectively, on a routine basis. Activities will be scheduled on an Event Timeline for execution, incorporated into a Detailed Activity Schedule (DAS) and ultimately uplinked to the spacecraft via an Absolute Time Command (ATC) load or Ground Script. See CCB Approval Process diagram 4.3-1.



When an Activity is created or modified, it must first be validated in a test environment to ensure integrity of command mnemonics and parameters. The FOT Configuration Control Board (CCB) will review and approve all Activities prior to incorporation into the Project Data Base (PDB) and use in an operational environment.

4.4 Responsibilities

It is the responsibility of each Instrument Team to define and maintain Activities required to perform scientific observations and any functional operation associated with their instrument. The FOT will be responsible for defining and maintaining all spacecraft subsystem Activities. The FOT is also responsible for the configuration of all Activities in the PDB and for implementing the CM process which governs these Activities.

4.5 Prerequisites/Constraints

Once subsystem Activities are identified, defined, tested and approved, they must be incorporated into the Project Data Base (PDB).

- ~~1.) Activities must be in compliance with system designed constraints and spacecraft requirements~~
- ~~1.) Commands and Command Procedures used in an Activity must exist in the PDB~~
- ~~1.) The nominal cycle for changes to be implemented will take approximately 3 days. During anomalous conditions, the review process will take approximately 6 hours~~
- 1.) Activities must be in compliance with system designed constraints and spacecraft requirements
- 2.) Commands and Command Procedures used in an Activity must exist in the PDB
- 3.) The nominal cycle for changes to be implemented will take approximately 3 days. During anomalous conditions, the review process will take approximately 6 hours

4.6 Procedure

4.6.1 Modify or Create an Activity

FOT and IOT members can create new Activities or make modifications to existing ones via the Activity Definer tool in the Planning and Scheduling (PAS) software.

4.6.2 Validation Process

The validation process takes place in a test environment, which “mirrors” the operational environment. All candidate Activities are checked for:

- ~~1.) Valid Syntax~~
- ~~1.) Valid Ranges~~
- ~~1.) Valid commands in the database~~
- ~~1.) Valid parameters~~
- 1.) Checks against defined constraints in the database Valid Syntax
- 2.) Valid Ranges
- 3.) Valid commands in the database
- 4.) Valid parameters
- 5.) Checks against defined constraints in the database

4.6.3 Timeline Scheduling

Once an Activity has been validated, it is scheduled onto a “What-if” or Master plan on an Event Timeline. The PAS software will automatically perform an Activity Level constraint check. Any “Soft” or “Hard” constraints must be handled accordingly (see Constraints SOP).

Hard constraints are considered hazardous to spacecraft health and safety and will not be incorporated into an ATC load or Ground Script. Soft constraints serve as warnings, but require a waiver by the cognizant party if they are deemed harmless to instrument or spacecraft operations. Any Hard constraint must be resolved prior to creating a Detailed Activity Schedule (DAS).

4.6.4 Detailed Activity Schedule (DAS) Creation

An Activity is expanded into a Command List and included in the Detailed Activity Schedule (DAS) when Load Generation is performed. At this point, the Activity being scheduled is constraint checked at both the Activity Level and the Command Level.

4.6.5 Final Product Output

Successful completion will yield a constraint-free ATC load and Ground Script reflecting the scheduled ~~Activity(s)- Activity(s)~~.

- 1.) ATC Load - Contains absolute time-tagged commands for both instrument and spacecraft subsystems which execute on-board the spacecraft when it is uplinked into the Absolute Time Processor.
- 2.) Ground Script - Used by the FOT Command Controllers to execute routine, pre-planned and real-time commanding of spacecraft, instrument, and ground system configurations.

4.6.6 Submit CCR Form

The user requesting the implementation of a new or modified Activity must submit a Configuration Change Request (CCR) form along with the test results. In the CCR, the user must describe all changes to the Activity and provide a reason for the changes. See Appendix 4.8.1.

4.6.7 Test Results / Final PDB Configuration

- 1.) Test results will be provided to the CCB panel in the form of reports upon successful completion of the validation procedures mentioned above.
- 2.) The user requesting implementation of an Activity into the PDB will fill out and submit a Validation Form.
- 3.) The FOT CCB panel will review all relevant changes pertaining to the CCR package.
- 4.) ~~If the CCR is approved, it will be configured into the PDB. If it is rejected, an explanation will be provided to the originator of the Activity. The new Activity with modifications is then resubmitted to the CCB. See Configuration Management SOP. If the CCR is approved, it will be configured into the PDB. If it is rejected, an explanation will be provided to the originator of the Activity. The new Activity with modifications is then resubmitted to the CCB. See Configuration Management SOP.~~

4.7 References

Release A & B FOS Operations Scenarios 605-CD-003-001

Flight Operations Segment (FOS) Operations Tool Manual for the ECS Project 609-CD-005-002

Operations Configured Items (CI) Document 706-CD-002-001

Configuration Management SOP

Constraints SOP

4.8 Appendix

N/A

4.8.1 ~~Example Configuration Change Request (CCR) Form~~

~~FOTAM-1 CONFIGURATION CHANGE REQUEST~~

~~Requested by:~~ Ed Rodriguez ~~CCR Number:~~

~~CCR Date:~~

~~CI Type:~~ SOP

~~CI ID:~~ Activity Definition

~~Revision:~~ N/A

~~Subsystem:~~ Ground

~~Document Traceability:~~

~~N/A~~

~~Reason for Change:~~

~~The purpose of this Standard Operating Procedure (SOP) is to document the procedure for creating and/or modifying Activity Definitions to support both instrument and spacecraft operations. This SOP provides the user with a description of what Activity Definitions are, and how they are created, approved and incorporated into the Project Data Base (PDB).~~

	Initials	Date	Action
CCR Originator:	EJR	7/28/97	
FOT Ops Man.:			
FOT S/C Man.:			
NASA FOD:			
CCB Chair:			Accept / Reject (circle one)

Comments:

5. Baseline Activity Profile (BAP) SOP

5.1 Originator

5.1.1 Name: Ed Rodriguez Date: 4/28/97

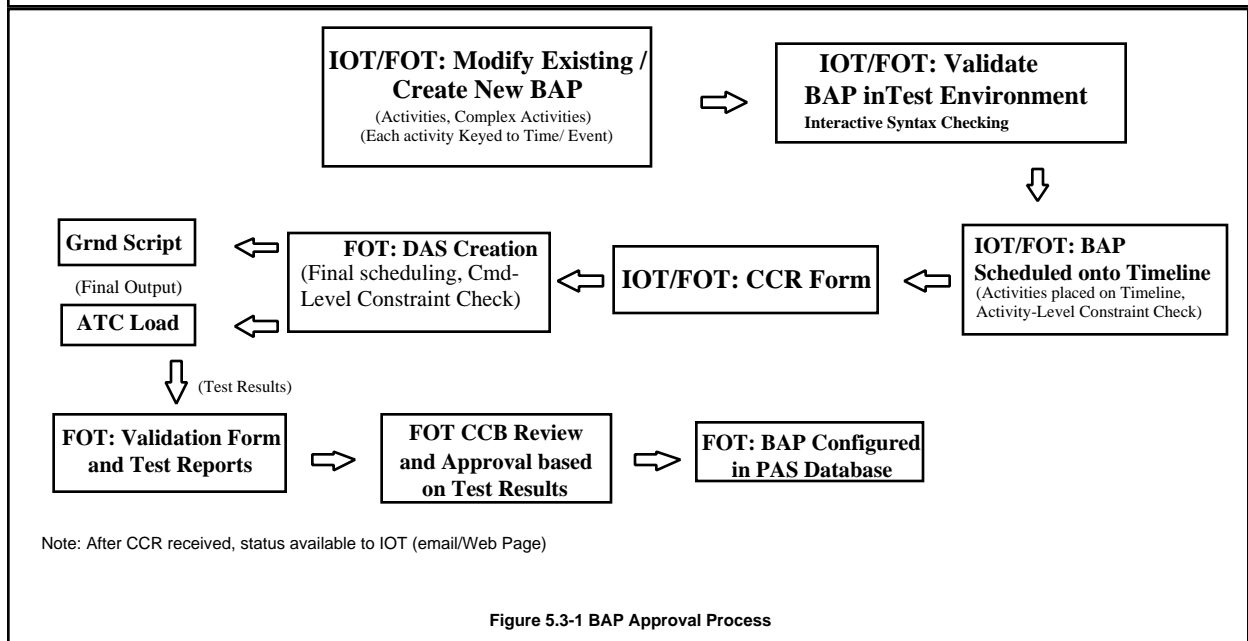
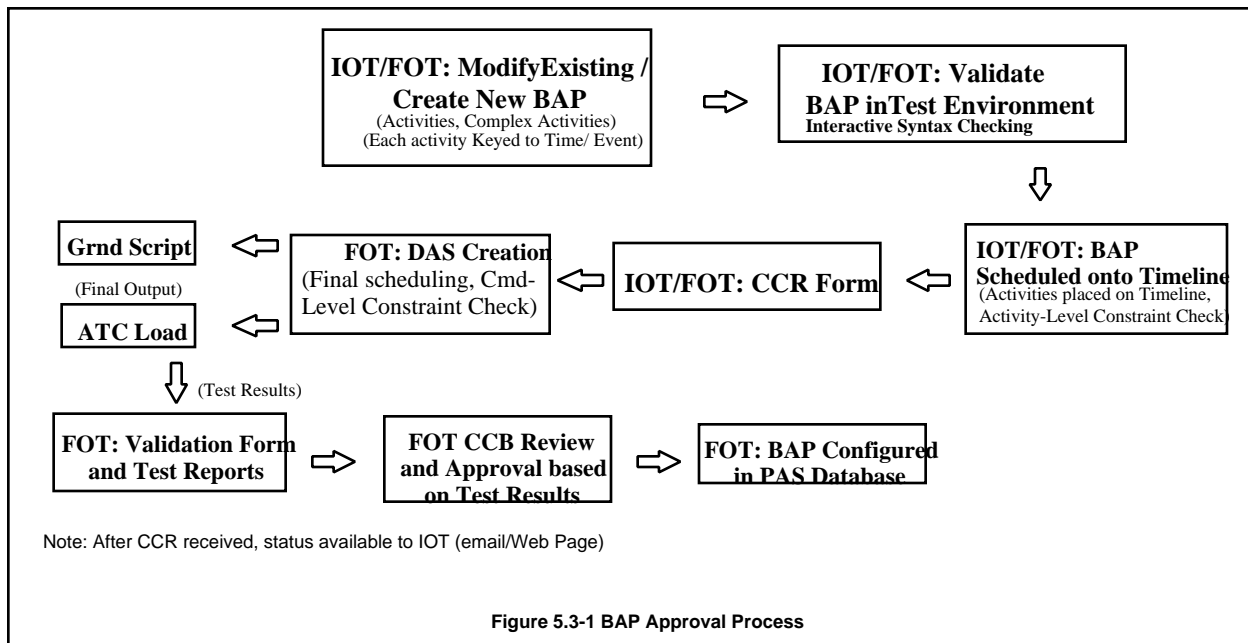
5.1.2 Revised by: Ed Rodriguez Date: 7/29/97

5.2 Purpose

The purpose of this Standard Operating Procedure (SOP) is to document the procedure for developing and configuring Baseline Activity Profiles (BAP's) supporting both instrument and spacecraft operations. This SOP provides the user with a description of how BAP's are created, approved, modified and incorporated into the Planning and Scheduling (PAS) Operational Database.

5.3 Background

The Flight Operations Segment (FOS) Planning and Scheduling subsystem (PAS) provides users with a BAP Definer tool to create or modify Baseline Activity Profiles. BAP's consist of numerous activities which are scheduled to support routine or normal instrument and spacecraft operations. These activities can be triggered to execute based on space network/ground network events (Real-time contacts), orbital events or user specified times. A BAP is a file which encompasses pre-defined activities with both Absolute Time Commands (ATC) and Ground Commands to be executed on a repetitive basis. Once the user defines the operation desired, it is scheduled to execute repeatedly for as long as Flight Dynamics data is available. BAP's are building blocks for the Detailed Activity Schedule. (See BAP Approval Process diagram 5.3-1)



5.4 Responsibilities

It is the responsibility of each Instrument Operations Team (IOT) to define, develop and schedule BAP's supporting their respective instrument on the Timeline. It will be the FOT's responsibility to define, develop and schedule spacecraft BAP's for all subsystems, and generate daily ATC loads based on the inputs provided by all teams.

5.5 Prerequisites/Constraints

- 1.) All Activities, Commands, ECL Directives and Command Procedures called out from a BAP must be defined in the Project Data Base (PDB).
- 2.) All BAP's must be able to be scheduled on the Timeline, the DAS, ATC Load and/or the Ground Script.
- 3.) All BAP's must be constraint checked and be in compliance with spacecraft system design limitations.
- 4.) All BAP's must be tested, validated and approved by an FOT CCB prior to being used in an operational environment.
- 5.) ~~The nominal cycle for changes to be implemented will take approximately 3 days. During anomalous conditions, the review process will take approximately 6 hours.~~ The nominal cycle for changes to be implemented will take approximately 3 days. During anomalous conditions, the review process will take approximately 6 hours.

5.6 Procedure

5.6.1 Modify / Create BAP

The Flight Operations and Instrument Teams can create or modify BAP's using the BAP Definer Tool in the Planning and Scheduling (PAS) software.

5.6.2 Validate BAP in Test Environment

All candidate BAP's will be checked out and validated in the test environment to ensure integrity. The PAS software contains a Validation Tool which checks the contents of the BAP for syntax, valid parameters, etc.

5.6.3 Schedule BAP on Event Timeline

Subsequent to Validation, the BAP is scheduled or installed on the Event Timeline as a "What-If" or Master Plan. Activities and Commands within the BAP are ultimately expanded into a Command List and incorporated into the ATC load and Ground Script when Load Generation is performed. ~~The integrity of the BAP being used is verified by visual inspection and against the Spacecraft Simulator (SSIM).~~ The integrity of the BAP being used is verified by visual inspection and against the Spacecraft Simulator (SSIM). PAS software performs an Activity Level constraint check on the BAP. Any "Hard" constraint violation which appears must be resolved. "Soft" constraint violations can be incorporated into an ATC load with approval.

5.6.4 Create Detailed Activity Schedule (DAS)

A BAP is expanded into a Command List and included in the Detailed Activity Schedule when Load Generation is performed. The scheduled BAP will be constraint checked at both the Activity Level and the Command Level during this process.

5.6.5 Final Product Output

Successful completion of the Load Generation process will yield a constraint-free ATC load and Ground Script reflecting the scheduled Activities within the BAP.

1. ATC Load - Contains absolute time-tagged commands for both instrument and spacecraft subsystems which execute on-board the spacecraft when it is uplinked into the Absolute Time Processor.
2. Ground Script - Used by the FOT Command Controllers to execute routine, pre-planned and real-time commands for the spacecraft, instrument, as well as performing ground configurations.

5.6.6 Submit CCR Form

The user requesting the implementation of a new or modified BAP must submit a Configuration Change Request (CCR) form along with the test results. In the CCR, the user must describe all changes to the BAP and provide a reason for the changes. See Appendix 5.8.1.

5.6.7 Test Results / Configuration into Operational Data Base

1. Test results will be provided to the FOT CCB in the form of reports upon successful completion of the validation procedures mentioned above.
2. The user requesting incorporation of a BAP into the Operational Database will fill out and submit a Validation Form.
3. ~~The FOT review panel will examine results from the BAP test procedure and accept or reject the proposed CCR. If approved, the BAP will be configured. If it is rejected, an explanation will be provided to the originator of the BAP. The new BAP with modifications is then resubmitted to the CCB. See Configuration Management SOP.~~ The FOT review panel will examine results from the BAP test procedure and accept or reject the proposed CCR. If approved, the BAP will be configured. If it is rejected, an explanation will be provided to the originator of the BAP. The new BAP with modifications is then resubmitted to the CCB. See Configuration Management SOP.

5.7 References

Release A & B FOS Operations Scenarios 605-CD-003-001

Flight Operations Segment (FOS) Operations Tool Manual for the ECS Project 609-CD-005-002

Release A and B Flight Operations Segment (FOS) Operations Scenarios Document 605-CD-003-001

Operations Configured Items (CI) Document 706-CD-002-001

Configuration Management SOP

Activity Definition SOP

5.8 Appendix

N/A

6. ECL Command Procedure SOP

6.1 Originator

6.1.1 Name: Nelson V. Pingitore Date: 4/3/97

6.1.2 Revised by: Nelson V. Pingitore Date: 4/29/97

6.2 Purpose

The purpose of this Standard Operating Procedure (SOP) is to document the process for creating and validating ECS Command Language (ECL) Command Procedures to support both instrument and spacecraft Real-time operations. This SOP provides the user with a description of the process for how ECL Command Procedures are created, validated, approved, and where they are used in Real-time operations. Although the details of the FOT Configuration Management (CM) process are covered in a CM-specific SOP, this ECL Command Procedure SOP will discuss aspects which are ECL Command Procedure-specific.

6.3 Background

ECL Command Procedures are comprised of collections of command mnemonics and ECL Directives. ECL Command Procedures are used to control the execution of multiple related commands during Real-time commanding of the spacecraft or instrument. Constructs shall be implemented within the procedure to test for pre-requisite conditions prior to actual commanding, control the commanding, and test for successful achievement of the desired end state. The Flight Operations Segment (FOS) provides users with an ECL Procedure Definer tool to create ECL Command Procedures. Both instrument and spacecraft operations teams will have the capability to generate, maintain, schedule, and submit requests for execution of ECL Command Procedures. ECL Command Procedures will be scheduled on the Planning and Scheduling (PAS) Timeline, incorporated into the Ground Script, or will be requested via the IST based Command Request Tool.

When an ECL Command Procedure is created, it will first be syntax checked via the FOS Procedure Builder Tool, then validated in a test environment with the AM-1 Spacecraft Simulator (SSIM), or via inspection, to ensure integrity of all command mnemonics and parameters. A Configuration Control Board (CCB) will review and approve all non-emergency ECL Command Procedures prior to installation into the EOC environment. Emergency ECL Command Procedures, prepared to address an ongoing emergency, are to be validated via the authors as time allows, given the time constraints of the emergency.

6.4 Responsibilities

It is the responsibility of each Instrument Operations Team (IOT) to define and maintain ECL Command Procedures required to perform any Real-time commanding associated with their instrument. The FOT will be responsible for defining and maintaining all spacecraft subsystem ECL Command Procedures. The FOT is also responsible for the configuration of all ECL

Command Procedures resident in the ECS Operations Center (EOC) including IOT procs, and for implementing the CCB process which governs these activities.

6.5 Prerequisites/Constraints

A requirement to execute a new spacecraft or instrument command function via an ECL Procedure or to update an existing one necessitates the creation of a new ECL Command Procedure or the modification of an existing one. Following the creation or modification of the ECL Command Procedure, its syntax should be checked via the FOS Procedure Builder Tool. If the ECL Command Procedure is Spacecraft bus related, then the next step is to execute it against the SSIM to demonstrate that it achieves the correct results. A Configuration Change Request (CCR) form is then filled out detailing the changes and the reasons for the change. The submission of the CCR form to the CCB initiates the process of approval of the modified ECL procedure and incorporation into the PDB (See Configuration Management SOP). New or modified CL procedures must include a CCB review and sign-off by all cognizant parties. This CCB review process will include the author, the FOT Operations Manager, Spacecraft Manager, and the NASA Flight Operations Director. Subsequent to this review, the CCB Chair will provide final sign-off to accept the Command Procedure as a configured change to the previously existing operational baseline. The nominal cycle for changes to be implemented will take approximately 3 days. During Spacecraft emergency conditions, the review process will be expedited as required to ensure S/C safety. See ECL Command Procedure CCB Approval process diagram 6.5-1.

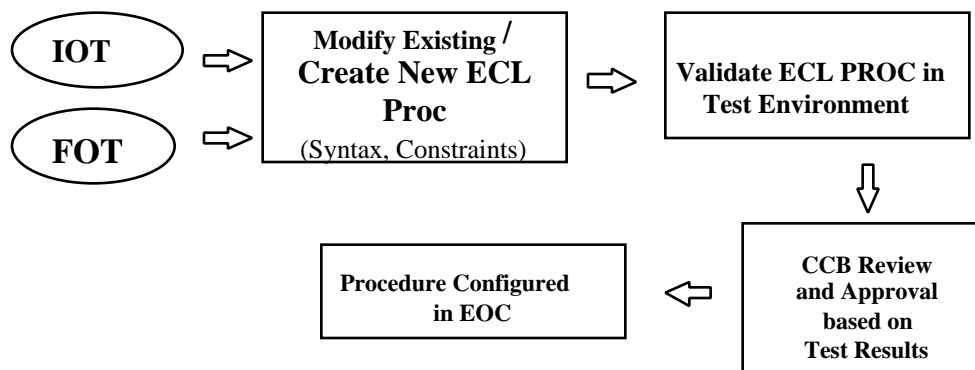


Figure 6.5-1: ECL Command Procedure CCB Approval Process

6.6 Process

1. The FOS provided Procedure Builder Tool (Reference #2) should be used to create a new, or modify an existing, ECL Command Procedure. Syntax and Constraint Checks are performed.
2. ECL Command Procedures, written by an FOT member in the EOC, are migrated to the FOS "Test" environment.

3. The ECL Command Procedure is executed against the SSIM to validate correct functional commanding.
4. For Instrument-related ECL Command Procedures, or subsystems not modeled by the SSIM, visual inspection will be performed to validate correct command functionality.
5. The ECL Command Procedure author submits the ECL Command Procedure to the FOT CCB (FOT Operations Manager, Spacecraft Manager, NASA Flight Operations Director) for review and approval.
6. Results of the CCB will be delivered to the ECL Command Procedure author.
7. Upon approval, the ECL Command Procedure will be installed as a Configured Item(CI) within the FOS for Real-time execution by the FOT.

6.7 References

1. Release A & B FOS Operations Scenarios 605-CD-003-001
2. Flight Operations Segment (FOS) Operations Tool Manual for the ECS Project 609-CD-005-002
3. ECS FOT Configuration Management SOP

6.8 Appendix

1. ECL Command Procedures Required Elements (TBS).

7. Display Builder SOP

71 Originator

7.1.1 Name: Alan D. Lampe Date: 3/25/97

7.1.2 Revised by: Ken McKenzie Date: 4/28/97

Completion required for delivery.

72 Purpose

The purpose of this Display Builder Standard Operating Procedure (SOP) is to document the manner in which display pages are created, utilized and maintained in accordance with established CM procedures.

73 Background

The Flight Operations Team/Instrument Operations Teams (FOT/IOT) create display pages in order to display telemetry points, ground system global parameters and Space Network (SN) provided User Performance Data (UPD). UPD is not provided to the IOT facility. These values are displayed during real-time events, simulations or replay events and are utilized to assess the health and safety of the spacecraft. One page can display multiple parameters from multiple sources. Multiple display pages can be grouped to form a room. Rooms are discussed in the Room Builder SOP.

Display pages are constructed through a top down approach. A separate display page will be created for select parameters from the entire spacecraft, for specific subsystems, for each instrument and for specific components for each subsystem or instrument.

74 Responsibilities

It is the responsibility of each FOT/IOT member that will be involved with the monitoring of telemetry to know how to create a page using the Display Builder. The FOT will be responsible for creating display pages for monitoring all aspects of the AM-1 spacecraft. IOT's will be responsible for creating display pages for monitoring their respective instrument. The FOT will be responsible the Configuration management (CM) process for display pages built for FOS-wide use.

75 Prerequisites/Constraints

A parameter must be defined in the Project Data Base (PDB) before it can be used in a display page. All EOC display pages are required to contain the following header information: **TBD**. The FOT will construct display pages for every telemetry parameter in the PDB. IOT's are required to submit display pages which include all available Instrument telemetry mnemonics. This will ensure that anomalous conditions can be investigated utilizing common telemetry displays.

76 Procedures

7.6.1 Display Page Elements

Display pages are defined by Users utilizing the FOS supplied Display Builder tool. Display pages are minimally composed of telemetry mnemonics or labels and the mnemonic value. Optionally, a display page can include any associated units, data quality and status flags. The mnemonics are displayed as defined in the PDB, or the User may choose to define a more descriptive label. The parameter value can be displayed as raw telemetry bits, decoded or converted values. The values can be displayed in Hex, Octal, Binary or Formatted. The units may include amps, volts, Fahrenheit, Celsius, radians, degrees, etc. The available flags are for limits, deltas, quality and static. The limits flags are PDB defined as red high, red low, yellow high and yellow low. The delta limits are PDB defined and indicate to the User if the parameter value violates the delta limit between successive updates.

7.6.2 Display_Builder

The Display_Builder actually is composed of 5 separate windows as follows:

- 1.) Display BuilderPalette - Used for positioning and selection of various page features.
- 2.) Display Item Format - Used to precisely position various items from the Builder Palette along the X and Y axis. Can also be used to vary the height and width of display fields and separators.
- 3.) Display Item Data Source - Used to specify desired mnemonic for placement on display page.
- 4.) Display Builder Console - Acts as a type of events page for functions related to the display builder.
- 5.) Display Builder - the area where page features and telemetry mnemonics are actually placed in order to create the display page.

7.6.3 Display Naming Convention

The FOT naming convention required for a display page utilized in the EOC is as follows:

- 1) The first three characters will indicate primary subsystem or instrument as defined in the PDB DFCD.
- 2) The next set of characters indicate a subsystem component or descriptor.
- 3) The last four characters are DISP to identify the file as a page display.

For example: CDH_SBT1_DISP is a Command and Data Handling Page with telemetry mnemonics for S-band Transponder number one.

7.6.4 Local versus CM builds

The FOS software enables Users to build display definitions for individual or system wide use. If the user elects to keep a display page for his/her own use, no EOC configuration management approval is required. The User can use any naming convention and will build the display with the

“Build to Local” option after creating the display. This page will only be available for the creator to invoke when connected to a string in the “tailored mode”. In order for a display page to be used by any system user, the page will be subjected to the FOT CM process and must be built with the “Build to CM” option after creation. Display pages must be “built to CM” if the User expects the FOT to monitor the page or a specific telemetry point on the display. CM built pages are available when the user is connected in the “tailored” or “mirrored” mode. If the User is connected in the tailored mode, the display software will access the “Local” version, and if connected in the mirrored mode, the display software will access the “CM” version.

7.6.5 Display Page Features

The following are color coded features for all display pages:

- 1) A parameter displayed in white indicates a static value.
- 2) A parameter displayed in blue indicates no limits defined in the PDB or the limits are turned off for the parameter.
- 3) A parameter displayed in green indicates the parameter has limits defined and the parameter is within those limits.
- 4) A parameter displayed in yellow indicates a high or low yellow limit violation.
- 5) A parameter displayed in red indicates a high or low red limit violation.

Display page parameters can be highlighted during a Real-time pass and used to create a real-time plot or spreadsheet.

7.6.6 CM process

The display page creator will complete a Configuration Change Request (CCR) form, attach a copy of the display page and submit for FOT CCB approval. CCB results will be delivered to the author. Upon approval, the display page will be installed as a Configured item within the FOS.

77 References

- _ FOS Operations Tools Manual for the ECS Project 609-CD-005-002
- _ FOS Integration and Testing Procedures
- _ FOS Critical Design Review
- _ FOT Configuration Management SOP

78 Appendix

N/A

8. Relative Time Command Sequences (RTCS) SOP

81 Originator

8.1.1 Name: William Muscovich Date: 9/12/97

8.1.2 Revised by: Date:

82 Purpose

The purpose of this Standard Operating Procedure (SOP) is to document the procedure for creating, testing, and uplinking a Relative Time Command Sequence (RTCS). This SOP will describe the process to be followed for submitting the proposed RTCS, validating it, and finally uplinking and activating it. It will also describe the responsibilities of the parties involved in implementing RTCS changes.

83 Background

Relative Time Command Sequences are used onboard the spacecraft to send a predefined sequence of commands to the spacecraft subsystems. Up to 128 different RTCSs can be defined in the AM-1 Spacecraft Control Computer (SCC) FSW Table #12. Each RTCS occupies 177 words of SCC memory. Each RTCS contains a one memory word header and sixteen command slots, which each occupy 11 memory words. The RTCS header word contains the Inhibit Group ID (0-255) associated with the RTCS and the number of commands (1-16) contained by the RTCS. Each 11 word command slot contains a one word time delay of up to 46 days, 14 hrs., 27 min., 50 sec. in EOS seconds (0-65535), a one word Command Word (CMDWD), a one word Command Descriptor (CMDDESC), and eight Command Data Words (DATAWDS).

RTCSs may contain up to sixteen commands, as defined within the AM-1 PDB. Up to (TBD) consecutive commands may contain a time delay of zero EOS seconds, to be dispatched at maximum speed allowed by the FSW Executive. The memory space occupied by the RTCS header word and the commands it contains must be contiguous. Only contiguous RTCSs (adjacent RTCSs within FSW Table #12) can be loaded within a single uplink load. RTCSs can be activated, reset, halted, disabled, or enabled by command. RTCSs can be commanded by stored commands (ATCs, RTCSs, TMONs) or by Real-time command. For more information about RTCS design please refer to (TBS).

84 Responsibilities

8.4.1 FOT/IOT Responsibilities

The FOT will allocate ranges of RTCSs to each of the instruments and the spacecraft bus. The FOT/IOT is responsible for submitting the RTCS to the Operations CCB for initial approval. The FOT/IOT is responsible for creating and managing RTCSs using the FOS RTCS Load Builder. The FOT will manage the onboard Inhibit ID's that are associated with each RTCS. The FOT will validate the RTCS by loading and testing on SSIM. After the FOT validates the

RTCSs it will submit the RTCSs to the Operations CCB for approval. The FOT is responsible for the uplink of the RTCS.

85 Prerequisites/Constraints

The following activities must be completed prior to uplinking an RTCS.

1. The RTCS must be validated by the FOT/IOT.
2. The Uplink procedure must be converted to ECL.
3. The RTCS must be reviewed and approved by the FOT Operations CCB.
4. The RTCS uplink procedure and load file(s) must be populated to the appropriate computer systems and directories.

86 Procedures

8.6.1 Creation of an RTCS

RTCSs will be created by FOT/IOT personnel using the FOS RTCS Load Builder. These RTCSs will be created from procedures or flowcharts designed by the FOT, IOTs, or VF. All output files are to be stored in a non-working directory and used for testing only until receiving CCB final approval.

8.6.2 RTCS Submittal

RTCSs can be submitted by the FOT or any of the Instrument Operations Teams (IOTs). The form that will be used for RTCS submittal is a memo containing the following information:

1. The reason for submitting this particular RTCS.
2. A date, if necessary, that the RTCS will have to be operational by.
3. An analysis of the spacecraft conditions required at the time of RTCS uplink and through RTCS enabling. (i.e. There can be no concurrent commanding of the RTCS external to the uplink/enable procedure by real-time commands, ATCs, RTCSs, TMONs.)
4. An analysis of the impact of the RTCS commands on other spacecraft systems. The description shall include a command by command impact analysis and an impact analysis of the RTCS as a whole. (i.e. will these commands increase or decrease the power load, data rates, etc.?).
5. An RTCS Association Description Table. (ref. sect. 12.8.1)
6. A descriptive table, or equivalent, that describes the RTCS. (ref. sect. 12.8.2)

This memo will be submitted to the Operations CCB. See appendix for details.

8.6.3 FOT Validation of the RTCS

The FOT will validate the RTCS using SSIM.

8.6.4 CCB Approval

The RTCS author will submit the validated RTCS to the Operations CCB. The FOT will submit the ECL procedure that will be used to uplink the RTCS. The CCB must approve the RTCS before it is uplinked.

8.6.5 Uplink of the RTCS

After the RTCS has been approved by the Operations CCB it can be uplinked to the spacecraft using the ECL procedure that was provided by the FOT. The RTCS will be uplinked with the RTCS in the disabled state to ensure that it can not be activated inadvertently. Uplink will be performed at a time deemed proper by the Flight Systems Engineer, and the RTCS author.

8.6.6 Enabling of the RTCS

After uplink, the RTCS will be enabled. The RTCS will be enabled at a time deemed proper by the Flight Systems Engineer, and the RTCS author.

87 References

1. FOS Operations Tools Manual for the ECS Project 609-CD-005-002
2. FOS Integration and Testing Procedures
3. FOT Configuration Management SOP
4. ICD between ECS and FSW Maintenance
5. EOS AM-1 MOR Review
6. EOS AM-1 Instruments Operations Workshop

88 Appendix

1. RTCS Association Description Table
2. RTCS Contents Description Table

881 RTCS Association Description Table

The table shall include a list and description of all Planning and Scheduling (PAS) Activities, RTCSs and TMONs which command the RTCS.

RTCS Association Description Table			
Associated P&S Activity(s):			
<u>Activity</u>	<u>Subsystem</u>	<u>Command</u>	<u>Command Description</u>
TBS	TBS	Mnemonic-1, Submnemonics...	Text command description.
	TBS	Mnemonic-2, Submnemonics...	Text command description.
	.	.	.
	.	.	.
	.	.	.
	TBS	Mnemonic-N-1, Submnemonics...	Text command description.
	TBS	Mnemonic-N, Submnemonics...	Text command description.
Associated RTCS(s):			
<u>RTCS #</u>	<u>Subsystem</u>	<u>ATC Command</u>	<u>Command Description</u>
TBS	TBS	Mnemonic-1, Submnemonics...	Text command description.
	TBS	Mnemonic-2, Submnemonics...	Text command description.
	.	.	.
	.	.	.
	.	.	.
	TBS	Mnemonic-14, Submnemonics...	Text command description.
	TBS	Mnemonic-16, Submnemonics...	Text command description.
Associated TMON(s):			
<u>TMON #</u>	<u>INHID</u>	<u>TMON Command</u>	<u>Command Description</u>
TBS	GRP_INHIB	n/a	n/a
	INHIB1	Mnemonic-1, Submnemonics...	Text command description.
	INHIB2	Mnemonic-2, Submnemonics...	Text command description.

882 RTCS Contents Description Table

Describe the RTCS contents, including Associated Inhibit ID, command count, and a full description of each command. The FOS RTCS Builder Tool includes this information in the RTCS Load Report.

RTCS Contents Description Table					
RTCS Header:					
<u>Inhibit ID</u>	<u>Command Count</u>				
(0 - 255)	(1 - 16)				
RTCS Commands:					
Command	Critical	Time Delay ¹	Command	Command Description	
1	y/n	(0-65535)	Mnemonic, Submnemonics...	Text command description.	
2	y/n	(0-65535)	Mnemonic, Submnemonics...	Text command description.	
3	y/n	(0-65535)	Mnemonic, Submnemonics...	Text command description.	
4	y/n	(0-65535)	Mnemonic, Submnemonics...	Text command description.	
5	y/n	(0-65535)	Mnemonic, Submnemonics...	Text command description.	
6	y/n	(0-65535)	Mnemonic, Submnemonics...	Text command description.	
7	y/n	(0-65535)	Mnemonic, Submnemonics...	Text command description.	
8	y/n	(0-65535)	Mnemonic, Submnemonics...	Text command description.	
9	y/n	(0-65535)	Mnemonic, Submnemonics...	Text command description.	
10	y/n	(0-65535)	Mnemonic, Submnemonics...	Text command description.	
11	y/n	(0-65535)	Mnemonic, Submnemonics...	Text command description.	
12	y/n	(0-65535)	Mnemonic, Submnemonics...	Text command description.	
13	y/n	(0-65535)	Mnemonic, Submnemonics...	Text command description.	
14	y/n	(0-65535)	Mnemonic, Submnemonics...	Text command description.	
15	y/n	(0-65535)	Mnemonic, Submnemonics...	Text command description.	
16	y/n	(0-65535)	Mnemonic, Submnemonics...	Text command description.	
¹ Units are EOS seconds. Maximum consecutive Time Delays with a value of zero is TBD.					

9. Derived Parameter SOP

91 Originator

9.1.1 Name: Kevin T. Work Date: 8/8/97

9.1.2 Revised by: Date:

92 Purpose

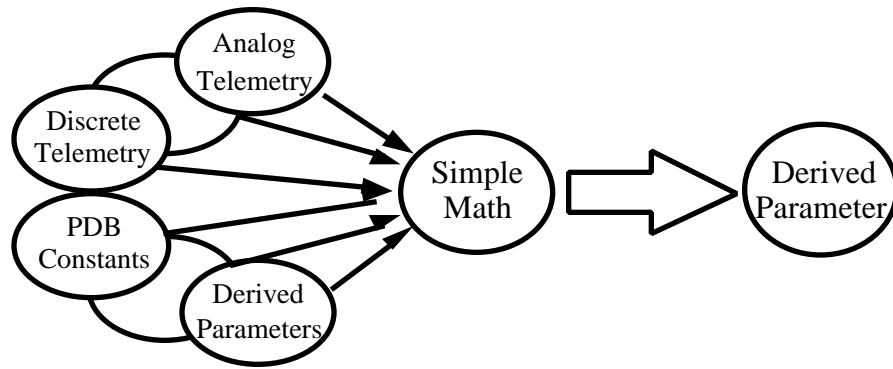
The purpose of the Derived Parameter Standard Operations Procedure (SOP) is to document the manner in which new or modified parameters are created, approved, and incorporated into the operational system in accordance with established FOT configuration management (CM) policies. These Derived Parameters are used by both instrument (IOT) and spacecraft operations teams (FOT) to aid in realtime and post pass analysis of telemetry.

93 Background

All AM-1 spacecraft telemetry is defined in the Project DataBase (PDB). Contained in the telemetry parameter section of the PDB are packet and parameter specifications, telemetry descriptions, engineering unit conversions including all necessary polynomial coefficients, multiple red/yellow limit sets, delta limits, and discrete state specifications. Any of the telemetry parameters defined in the PDB can be viewed during a realtime contact or extracted from the telemetry archive for replays, reports, tables, plots, or trending.

In addition to the available spacecraft telemetry parameters, simple mathematical and logical combinations of parameters are also available for realtime and post pass analysis. These combinations are referred to as Derived Parameters and are defined in the telemetry section of PDB. Currently, up to six input parameters can be combined using a simple user defined equation with the result displayed in realtime. Along with the analog and discrete telemetry of the PDB, other previously defined Derived Parameters, or PDB defined constants can be incorporated in the equation. Each input analog telemetry parameter can be used pre or post engineering unit conversion. The computation rate of the Derived Parameter is also set in the PDB in units of seconds.

Once a Derived Parameter is defined in the PDB, the ground system treats it the same as any other spacecraft telemetry parameter. The newly developed parameter can be viewed in realtime on a telemetry display, checked by the Decision Support System (DSS), plotted in a report, and trended. All limit capabilities are also available. Derived Parameters are distinguishable from the other telemetry through a unique naming convention.



94 Responsibilities

It is the responsibility of the user to define the combination of inputs needed to create a Derived Parameter. The user will also develop and test the parameter while in the appropriate test environment. Both the FOT and IOTs have access to the test environment and only there are the necessary Database Utilities and the Test PDB available. After completion of validation, the user must submit a Configuration Change Request (CCR) form and related documentation to the FOT Configuration Control Board (CCB) for review and approval. The Database Administrator (DBA)/Configuration Manager will manage and implement CCB approved CCRs.

95 Prerequisites/Constraints

Desired inputs to the Derived Parameter must be defined in the appropriate sections of the PDB. Currently the number of input parameters per Derived Parameter is limited to six.

The number of Derived Parameters supported at a given time shall not exceed fifty.

Operators are evaluated in the order specified. No inherent order of precedence exists. Care must be taken when developing the parameter equations. Valid operators include: addition, subtraction, negation, multiplication, division, sine, arcsine, cosine, arccosine, tangent, arctangent, equal, not equal, less than, less than or equal, greater than, greater than or equal, logical AND, logical OR, and logical NOT.

When creating new Derived Parameters, the following naming convention must be used. The first three letters of the Derived Parameter are the subsystem designator followed by the sample type and then a **D** to designate the pseudo operation. The rest of the name is user choice with the total name limited to twenty characters. All characters must be capitalized. Refer to the Data Format Control Document for a listing of correct subsystem designators and sample types. An Example name follows: EPS_ID_TOTAL_SA_PWR. This convention is necessary to distinguish Derived Parameters from actual spacecraft telemetry.

96 Procedures

9.6.1 Complete a CCR Form

The user completes a CCR form according to policies developed in the FOT Configuration Management SOP. The CCB will direct the user to the appropriate test environment for Derived Parameter development and testing.

9.6.2 Develop the Derived Parameter

The user develops the equation used to calculate the Derived Parameter. Up to six previously defined input parameters are available for use in the equation. The order of the calculation is then manipulated to achieve the desired result.

The user must also specify the new Derived Parameter's desired name following the appropriate conventions stated above. The user and the DBA decide on a suitable computation rate, in seconds.

9.6.3 Update Project Database

The user then adds the new Derived Parameter to the PDB while in the test environment. The Derived Parameter mnemonic must also be added to the Telemetry Description Specification section of the PDB. Other items to enter include the desired units and the computation rate. If limit checking of the new parameter is desired, then the Red/Yellow Limit Specification needs to be updated. If a constant is used in the derivation, it must be defined in the Telemetry Constant Specification. Refer to section 10.4.4.4.3 letter K of the FOS Tools Manual for a description of the FOS Derived Telemetry Specification and other related utilities.

The following trace is used to reach the Derived Parameter generation/modification and related tools:

```
> DATABASE UTILITIES
    > DATABASE ACCESS
        > PDB PARAMETERS
            > TELEMETRY PARAMETERS
                > DERIVED TELEMETRY SPECIFICATION
                    (required)
                > TELEMETRY DESCRIPTION SPECIFICATION
                    (required)
                    ***** related utilities *****
                > RED/YELLOW LIMIT SPECIFICATION
                    (optional)
                > LIMIT SELECTION SPECIFICATION
                    (optional)
                > DELTA LIMIT SPECIFICATION
                    (optional)
                > TELEMETRY CONSTANT SPECIFICATION
                    (optional)
```

9.6.4 Validation

Once the new Derived Parameter has been added to the test PDB, verification can be accomplished in many ways. Because Derived Parameters are recalculated each time they are queried, archived telemetry can be used to test the new mnemonic. A Data Replay, an Analysis Request, or the simulator are all available for evaluation in the test environment. An Analysis Request of the Derived parameter and all the inputs over a interval of archived telemetry will provide the easiest validation. A Data Replay requires a special Telemetry Display to be generated and the simulator requires the user to have operation experience.

The user is responsible for ensuring the correct outcome of the equation. After completion of testing, the user completes a Validation form following the polices developed in the FOT Configuration Management SOP.

9.6.5 Configuration Management

The CCR and Validation forms along with a report on the Derived Parameter are submitted to the FOT CCB for review. If rejected, the package will be returned to the user with comments or questions. Approved Derived Parameters will be migrated to the operations area by the FOT DBA.

97 References

Data Format Control Document for EOS FOS (505-10-35)

FOT Configuration Management SOP

FOS Operations Tools Manual for the ECS Project (609-CD-005-002)

98 Appendix

N/A

10. Analysis Algorithm SOP

101 Originator

10.1.1 Name: Kevin T. Work Date: 7/25/97

10.1.2 Revised by: Date:

102 Purpose

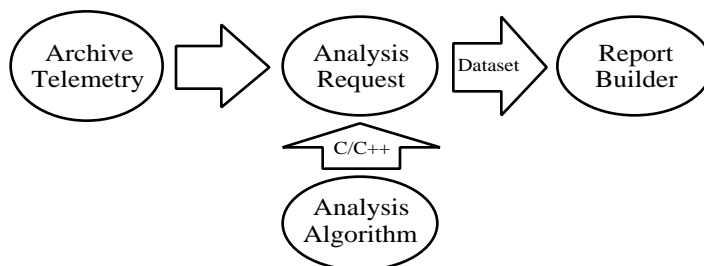
Analysis Algorithms are used by both instrument (IOT) and spacecraft operations teams (FOT) to aid in post pass analysis of telemetry. The purpose of the Analysis Algorithm Standard Operations Procedure (SOP) is to document the manner in which new or modified algorithms are created, approved, and incorporated into the operational system in accordance with established FOT configuration management (CM) policies.

103 Background

AM-1 spacecraft telemetry is archived within the ground system, parsed into hourly files by type and channel (see section 10.3 of the FOS Tools Manual). The Analysis Request Builder tool is used to extract desired telemetry parameters over a given time span and form a dataset or carryout file (see section 10.2 of the FOS Tools Manual). Numerous sampling methods for each parameter are available during this process including: sampling only upon a change, every Nth sample, or to extract all the data. Statistics such as minimum, maximum, mean, and standard deviation over specified periods up to one day is another option of the Analysis Request builder. The Report Generator tool is then used to browse, edit, or print dataset reports (see section 7.12 of the FOS Tools Manual). The reports can contain plots and tables of desired parameters. Formats of these plots and tables are established during the analysis request stage. For routine operations, the entire procedure can be placed into a Standing Order and execute on a user selectable interval.

In addition to the available spacecraft telemetry parameters, user defined C/C++ algorithms can manipulate available parameters and output new parameters to also be placed into reports, plots, and tables. Up to twenty input and twenty output variables can be processed per algorithm. See sample of user-defined algorithm written in C++ in Appendix 1.

The algorithms are dynamically linked to the Analysis Request process. The input variables of the algorithm are tied to available telemetry parameters and the output variables are assigned to new distinct parameter names following a FOT naming convention (refer to section 10.5.1). The timing of the algorithm is also specified during the analysis request. A timer is available as well as using another telemetry parameter, not necessarily an algorithm input variable, as a trigger.



104 Responsibilities

It is the responsibility of the user to define and develop each desired Analysis Algorithm. The FOT Configuration Control Board (CCB) will provide direction as to appropriate test environments and platform specific compiler flags. After completion of validation, the user must submit a Configuration Change Request (CCR) form and related documentation to the CCB for review and approval. The Database Administrator (DBA)/Configuration Manager will manage and implement CCB approved CCRs.

105 Prerequisites/Constraints

Desired input parameters to the Analysis Algorithm must be defined in the Project Data Base (PDB) or produced by another called algorithm.

The C/C++ Analysis Algorithm is limited to twenty input and twenty output variables. The algorithm must contain the required comments for input and output variables, including project specific data types. See example contained in Appendix 1 for other programming constraints.

FOS software can run on multiple platforms, and analysis requests can be farmed out to any of these platforms, therefore specific compiler flags are required.

Compiled C/C++ Analysis Algorithms must be registered to the FOS ground system. This is done using the Algorithm Registration tool. The FOS software will verify the format of the algorithm source file.

10.5.1 Analysis Algorithm Naming Convention

When linking the output variables of the algorithm to user defined output parameters to be included in reports, the following FOT naming convention must be used. The first three letters of the output parameter are a subsystem designation followed by ALG and then user choice. Example output parameter: **EPS_ALG**_Total_SA_Power. This is necessary to distinguish algorithm derived parameters from actual spacecraft telemetry. Variable names within the Analysis Algorithm are unrestricted.

106 Procedures

10.6.1 Complete CCR Form

The user completes a CCR form according to policies developed in the FOT Configuration Management SOP. The CCB will direct the user to the appropriate test environment for algorithm development and testing. At this time, the CCB will also notify the user of any platform specific compiler flags needed during the undertaking.

10.6.2 Develop Analysis Algorithm

The user develops an Analysis Algorithm as a C/C++ program. Up to twenty input variables per algorithm are available to calculate up to twenty output variables. Certain formats must be followed in order to interact with FOS software. Refer to the comments in the sample program located in Appendix 1.

10.6.3 Compile Analysis Algorithm

The Analysis Algorithm is compiled while in the test environment using any necessary compiler flags assigned by the CCB.

10.6.4 Register Analysis Algorithm

Before an algorithm can be used in an analysis request it first must be registered. The FOS Algorithm Registration tool is used to check for the proper format needed to interact with the FOS software. The registration process will return a success message or provide a list of errors.

10.6.5 Adding Algorithm to Analysis Request

The algorithms are accessed from the Analysis Request tool. The “Algorithm...” button brings up a list of available registered algorithms. After the desired algorithm is selected, the Input TLM and Output TLM must be assigned. The variables within the algorithm are linked to available input parameters and appropriate output parameters are created, following the correct naming convention (refer back to section 10.5.1), and linked to the algorithm’s output variables. The algorithm timing is also established at this point. Selected parameters can be used as triggers or a timer is also available.

10.6.6 Validation

The user is responsible for ensuring the correct outcome of the algorithm. A test Analysis Request, including the algorithm, is submitted in the appropriate test environment. The output of the Report Generator is then verified for proper algorithm execution. After completion of testing a Validation form must be completed following the policies developed in the FOT Configuration Management SOP. Included on the Validation form should be results of the compilation and registration steps in addition to test results.

10.6.7 Configuration Management

The CCR and Validation forms along with a report on the Analysis Algorithm are submitted to the FOT CCB for review. If rejected, the package will be returned to the user with comments or questions. Approved Analysis Algorithms will be migrated to the operations area by the FOT DBA.

107 References

FOS Operations Tools Manual for the ECS Project 609-CD-005-002

FOT Configuration Management SOP

108 Appendix

1. Sample User Defined Algorithm written in C++

**1081: Sample User-Defined Algorithm written in C++, with explanatory comments:
(Extracted from the FOSTools Manual, section 1022)**

```
#include "EcTypes.h" // include file used to define User Algorithm specific data types
/* the following variables map to input and output parameters (mnemonics) of the algorithm
    and MUST be global */
/*INPUTS #a required comment needed by the User Algorithm Process. Distinguishes
    input and output parameters */
EcTReal SolarArrayCurrent1; // an input. EcTReal is an algorithm data type for floating point
numbers
EcTReal SolarArrayCurrent2; // another input
EcTInt SolarArrayCount; // and integer data type input
//OUTPUTS # another required comment, indicating the remaining variables map to outputs
EcTReal TotalCurrent; // a floating point output
EcTInt AnInteger; // and integer output
EcTBoolean AnInteger_OutputComplete = EcTFalse; /* an output flag telling the algorithm
    software whether to output the value of AnInteger. Useful for
    performing running sums where the algorithm may execute every second
    but results are only needed every ten minutes. */
extern "C" UserFunction() /* algorithms must be extern "C", return void, and take no arguments.
    The function name must be the same as the filename minus extension ie.
    UserFunction.C would be the filename for this algorithm */
{
    static int count = 0; // local variables need not be EcTInt/EcTReal
    // compute total solar array current and store in output
    TotalCurrent = SolarArrayCurrent1 + SolarArrayCurrent2;
    // now perform a meaningless example of a running sum
    count += SolarArrayCount;
    if (count > SOME_BIG_NUMBER)
    {
        AnInteger = count;
        AnInteger_OutputComplete = EcDTrue; /* tell the software to output AnInteger */
        count = 0;
    }
    else
    {
        AnInteger_OutputComplete = EcDFalse; /* tell the software NOT to output AnInteger */
    }
}
```

11. Inhibit ID SOP

111 Originator

11.1.1 Name: William Muscovich Date: 9/12/97

11.1.2 Revised by: Date:

112 Purpose

The purpose of this Standard Operating Procedure (SOP) is to describe and document the procedure for initiating, submitting and managing AM-1 spacecraft Inhibit IDs (INHIDs). It also describes the responsibilities of the parties involved.

113 Background

Inhibit IDs are resident onboard the spacecraft to provide a means for inhibiting/allowing ATCs, RTCSs, and TMONs as desired. Inhibit IDs are a two part process. Part one is a central table of 255 INHIDs which can be “set” or “cleared” as desired. Part two resides in each specific ATC, RTCS, or TMON and determines which of the INHIDs is associated with the particular process. Part two is implemented within the EOS PDB, the FOS (RTCS Load Builder Tool or Configuration File - TBD), and VF Delivered TMON Builder Tool for ATCs, RTCSs, and TMONs respectively.

Inhibit IDs are numbered 1 - 255 and are stored in the SCC as (SCC Table #TBD). Each of the 255 INHIDs are set to TBD (1?) to indicate “set” (inhibit) or TBD (0?) to indicate “clear” (allow). Prior to executing an ATC or RTCS and prior to checking a TMON, the FSW reads in the INHID(s) associated with the ATC, RTCS, or TMON. Next it checks the associated INHID in the central table. If the associated INHID is “clear” the activity will be performed nominally. If the INHID is “set”, the activity will be inhibited. Each individual ATC command contains one INHID association field. Each whole RTCS group contains one INHID association field. No capability for INHID association of individual commands within an RTCS exists. Each TMON group contains three INHID association fields. One inhibits/allows the TMON group from being checked as a whole. The second inhibits/allows the dispatching of Command-1 for the TMON group. The third inhibits/allows the dispatching of Command-2 for the TMON group. For more information about INHID design please refer to (TBS).

114 Responsibilities

11.4.1 FOT/IOT Responsibilities

The FOT will allocate ranges of Inhibit IDs to each of the instruments and the spacecraft bus. The FOT/IOT is responsible for submitting the desired INHID implementation or change to the

Operations CCB for initial approval. The FOT is responsible for managing and monitoring INHIDs onboard the spacecraft and within the FOS ground system. The FOT will submit the INHIDs to the Operations CCB for final approval. The FOT is responsible for creating and executing the Command Procedure for uplinking the INHID. The implementation of and changes to ATC, RTCS, and TMON INHID association fields within the ATCs, RTCSs, and TMONs themselves are governed by their respective SOPs.

115 Prerequisites/Constraints

The following activities must be completed prior to uplinking an INHID.

1. The INHID implementation/change may be reviewed by FSW Maintenance. (TBD)
2. The Uplink procedure must be generated in ECL.
3. The INHID implementation/change must be reviewed and approved by the FOT Operations CCB.
4. The INHID uplink procedure and load file(s) must be populated to the appropriate computer systems and directories.

116 Procedures

11.6.1 INHID Submittal

INHIDs implementations/changes can be submitted by the FOT or any of the Instrument Operations Teams (IOTs). The form that will be used for INHID submittal is a memo containing the following information:

1. The reason for submitting this particular INHID.
2. A date, if necessary, that the INHID implementation/change must be operational by.
3. An analysis of any special spacecraft or other conditions required at the time of INHID implementation/change.
4. An analysis of the impact of the INHID implementation/change on other spacecraft systems. (i.e. will this change require modification of associated ATCs, RTCSs, or TMONS?)
5. An association description table listing and describing all ATCs, RTCSs and TMONs associated with the INHID. (ref. sect. 12.8.1)

This memo will be submitted to the Operations CCB. See appendix for details.

11.6.2 Creation of an INHID

Creation of INHIDs is TBD, pending analysis of AM-1 FSW implementation.

11.6.3 FOT Validation of the INHID

FOT will validate the INHID using SSIM if necessary.

11.6.4 CCB Approval

The requestor of the INHID implementation/change will submit the FOT validated INHID to the Operations CCB. The FOT will submit the ECL procedure that will be used to uplink the INHID. The CCB must approve the INHID before it is uplinked.

11.6.5 Uplink of the INHID

After the INHID has been approved by the Operations CCB it can be uplinked to the spacecraft using the converted ECL procedure that was provided by FSW Maintenance. A FSW Maintenance representative may (TBD) be present in the EOC during the INHID uplink. INHIDs will be uplinked with the INHID in the disabled state to ensure that it can not be activated inadvertently. Uplink will be performed at a time deemed proper by FSW Maintenance personnel, the Flight Systems Engineer, and the INHID author.

11.6.6 Enabling of the INHID

N/A

117 References

1. FOS Operations Tools Manual for the ECS Project 609-CD-005-002
2. FOS Integration and Testing Procedures
3. FOT Configuration Management SOP
4. ICD between ECS and FSW Maintenance
5. EOS AM-1 MOR Review
6. EOS AM-1 Instruments Operations Workshop

118 Appendix

1. INHID Association Description Table

11.8.1 INHID Association Description Table

The table shall include a list and description of all ATCs, RTCSs and TMONs associated with the INHID.

INHID Association Description Table			
Associated ATC(s):			
<u>Subsystem</u>	<u>ATC Command</u>	<u>Command Description</u>	
TBS	Mnemonic, Submnemonics...	Text command description.	
TBS	Mnemonic, Submnemonics...	Text command description.	
TBS	Mnemonic, Submnemonics...	Text command description.	
Associated RTCS(s):			
<u>RTCS #</u>	<u>Subsystem</u>	<u>ATC Command</u>	<u>Command Description</u>
TBS	TBS	Mnemonic-1, Submnemonics...	Text command description.
	TBS	Mnemonic-2, Submnemonics...	Text command description.
	.	.	.
	.	.	.
	.	.	.
	TBS	Mnemonic-15, Submnemonics...	Text command description.
	TBS	Mnemonic-16, Submnemonics...	Text command description.
Associated TMON(s):			
<u>TMON #</u>	<u>INHID</u>	<u>TMON Command</u>	<u>Command Description</u>
TBS	GRP_INHIB	n/a	n/a
	INHIB1	Mnemonic-1, Submnemonics...	Text command description.
	INHIB2	Mnemonic-2, Submnemonics...	Text command description.

12. Telemetry Monitor (TMON) Groups SOP

121 Originator

12.1.1 Name: John Teter Date: 8/11/97

12.1.2 Revised by: Date:

122 Purpose

The purpose of this Standard Operating Procedure (SOP) is to document the procedure for creating, testing, and uplinking a Telemetry Monitor (TMON). This SOP will describe the process to be followed for submitting the proposed TMON, validating it, and finally uplinking and activating it. It will also describe the responsibilities of the parties involved in implementing TMON changes.

123 Background

Telemetry Monitors are used onboard the spacecraft to monitor critical telemetry points and take action if the assigned limits are violated. TMONs can activate RTCs, send spacecraft commands, or send a message. There are 75 different TMON groups capable of being programmed on the AM-1 spacecraft. Each TMON is capable of sending up to two commands. TMON groups can operate in two different modes. The first mode monitors the telemetry but does not send any commands, and the second mode allows the TMON to issue the programmed commands. Both the monitoring mode and the commanding mode can be disabled.

TMONs can be composed of up to 4 subgroups each. TMONs can be designed to disable themselves, latch themselves, or continue operation (without resetting the Out Of Limit (OOL) counter) when an OOL condition is detected. If no OOL condition is detected the TMON will exit the current group with an In Limit Condition (INL) and continue execution with the next TMON Group. In addition to this, the execution time of each TMON can be chosen by its place in the Master cycle, and the interval between each execution can be specified up to a maximum of once per Master Cycle. For more information about TMON design please refer to (TBS).

124 Responsibilities

12.4.1 FOT/IOT Responsibilities

The FOT/IOT is responsible for submitting the TMON to the Operations CCB for initial approval. The FOT is responsible for creating and managing TMONs using the tool provided by FSW Maintenance. The FOT will manage the Inhibit ID's that are associated with any TMONs. The FOT will submit TMONs to FSW Maintenance for review. After FSW Maintenance validates the TMONs the FOT will submit the TMONs to the Operations CCB for approval. The FOT is responsible for the uplink of the TMONs with FSW Maintenance supervision.

12.4.2 FSW Maintenance Responsibilities

The FSW Maintenance personnel are responsible for validating the TMONs submitted by the FOT. They will also be responsible for creating the Command Procedure for uplinking the TMON. FSW Maintenance personnel will oversee the TMON uplink in the EOC.

125 Prerequisites/Constraints

The following activities must be completed prior to uplinking a TMON.

1. The TMON must have been validated by FSW Maintenance.
2. The Uplink procedure must have been converted to ECL.
3. The TMON must have been reviewed and approved by the FOT Operations CCB.

126 Procedures

12.6.1 TMON Submittal

TMONs can be submitted by the FOT or any of the Instrument Operations Teams (IOTs). The form that will be used for TMON submittal is a memo containing the following information:

1. A logical flow chart, or equivalent, that describes the TMON limit checking and commanding. Also, a TMON entry table that describes some of the higher level functions of the TMON in Table format.
2. The reason for submitting this particular TMON.
3. A date, if necessary, that the TMON will have to be operational by.
4. An analysis of the impact of the TMON commands on other spacecraft systems(i.e. will these commands increase or decrease the power load, data rates, etc.?).

This memo will be submitted to the Operations CCB.

12.6.2 Creation of a TMON

TMONs will be created by the FOT personnel using the tool that is provided by FSW Maintenance. These TMONs will be created from procedures or flowcharts designed by the FOT engineers and/or IOTs. The TMONs will be created so that they will not be active when they are uplinked to the spacecraft. This is to ensure that the TMON is operating correctly before it is allowed to command the spacecraft.

12.6.3 Transfer of the TMON to FSW Maintenance

The TMONs will be transferred to FSW Maintenance using the following method (TBS). The FOT will coordinate the transfer with FSW Maintenance.

12.6.4 FSW Maintenance Validation of the TMON

FSW Maintenance is responsible for validating the TMON that is submitted by the FOT. Once the TMON has been validated by FSW Maintenance, the FOT can proceed with submitting the

TMON to the Operations CCB. The FOT will have the capability of validating the TMON using SSIM if necessary.

12.6.5 CCB Approval

The FOT will submit the validated TMON to the Operations CCB after it is approved by FSW Maintenance. Along with the TMON the FOT should submit the converted ECL procedure that will be used to uplink the TMON. The CCB must approve the TMON before it is uplinked.

12.6.6 Uplink of the TMON

After the TMON has been approved by the Operations CCB it can be uplinked to the spacecraft using the converted ECL procedure that was provided by FSW Maintenance. A FSW Maintenance representative shall be present in the EOC during the TMON uplink.

12.6.7 Activation of the TMON

TMONs will be uplinked with both the limit checking and commanding disabled. After uplink the TMON's limit checking should be enabled. The TMON should be allowed to operate in monitor mode until it is determined that the logic is working satisfactorily. The author of the TMON and the Flight Systems Engineer will be responsible for making the determination as to whether the TMON is operating properly. After both of these individuals are satisfied that the TMON is working properly, commanding can be enabled.

127 References

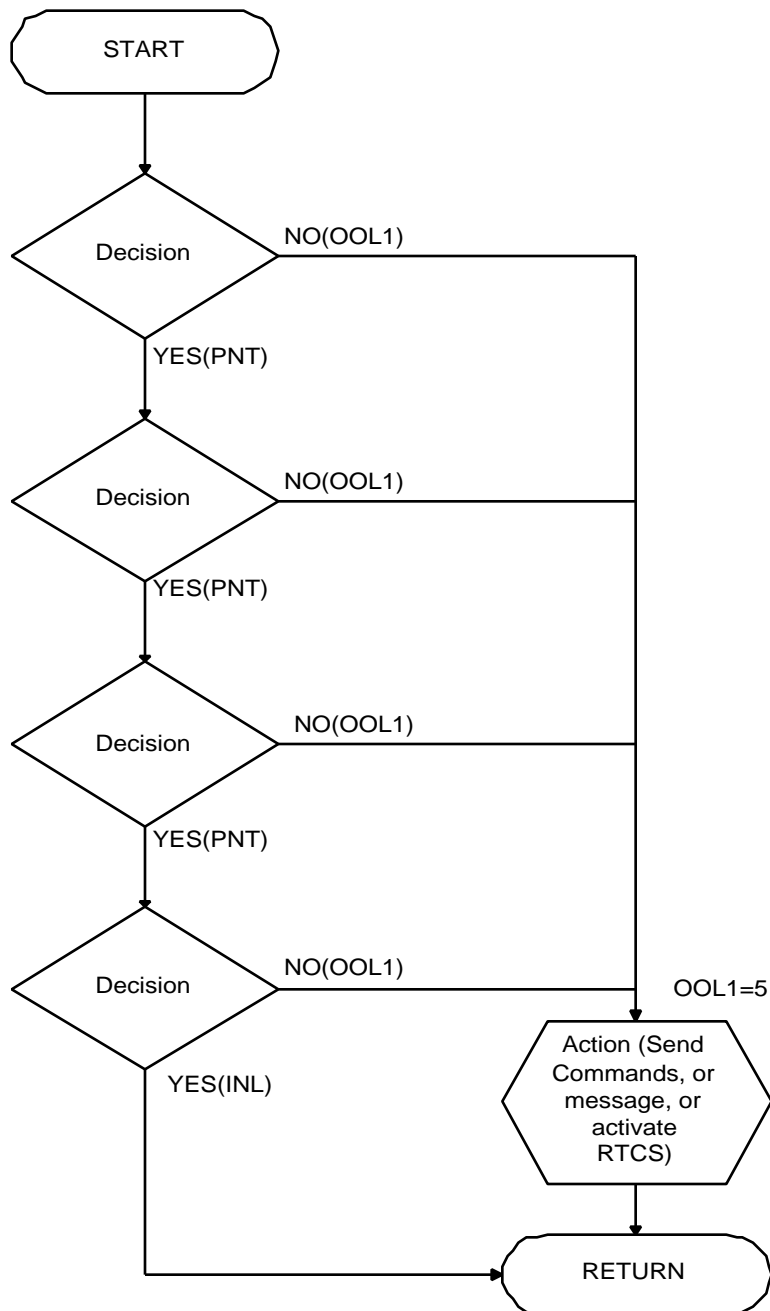
1. FOS Operations Tools Manual for the ECS Project 609-CD-005-002
2. FOS Integration and Testing Procedures
3. FOT Configuration Management SOP
4. ICD between ECS and FSW Maintenance
5. EOS AM-1 MOR Review
6. EOS AM-1 Instruments Operations Workshop
7. TMON Tool User's Guide (TBS)

128 Appendix

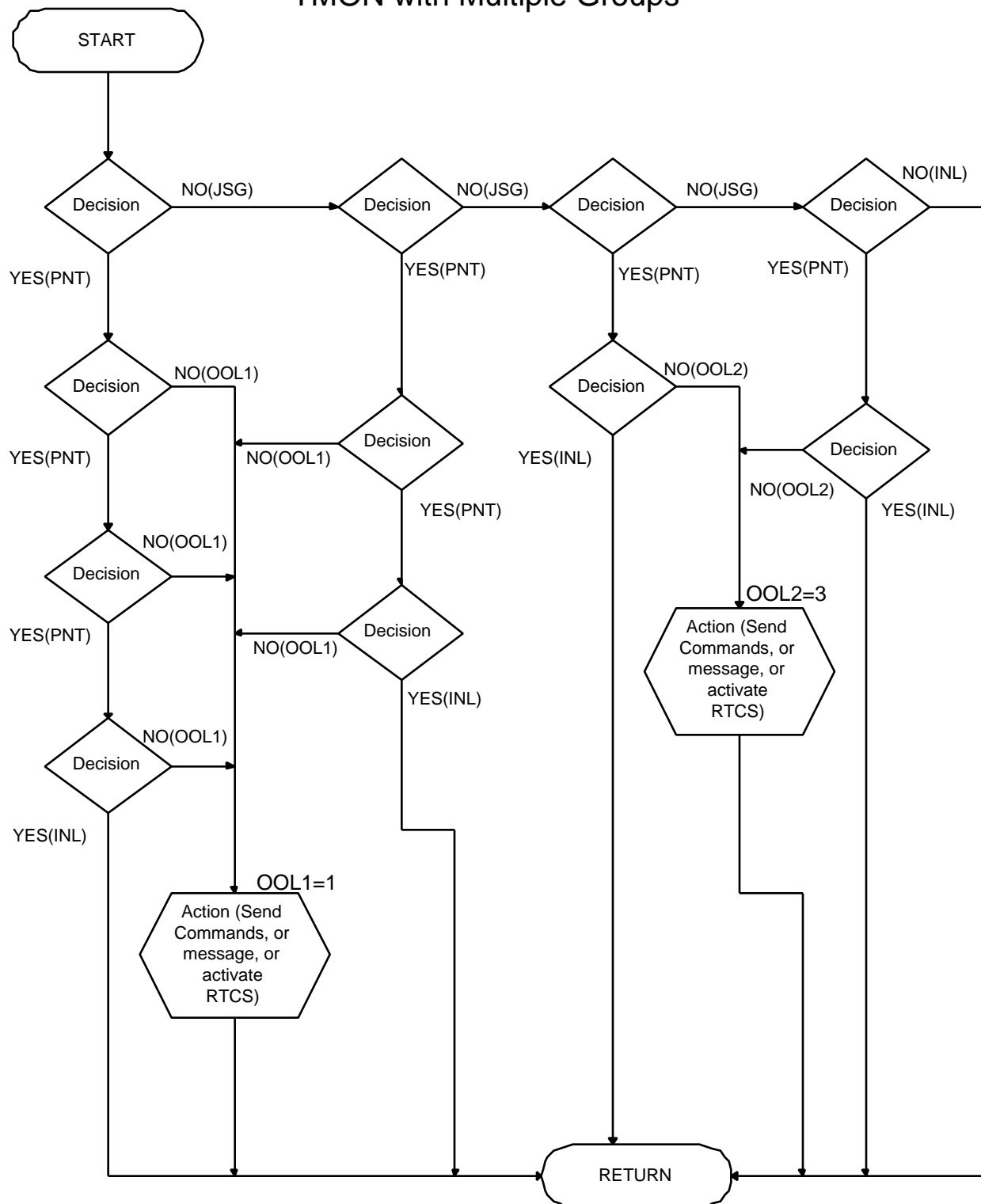
1. TMON Flow Chart Examples
2. TMON Entry Table
3. TMON Test Description Table

12&1 TMON Flow Chart Examples

TMON with Single group



TMON with Multiple Groups



1282 TMON Entry Table

TMON Group #	Group Input Controls			Group Output Controls				MAX OOL	Action When OOL1 reaches MAX		Action When OOL2 reaches MAX	
	TMON Group Inhibit ID	Executive Interval	Major Cycle Offset	After Trigger	Inhibit Bit Setting	Command Output	Log Report		Inhibit ID Status Bit to SET	Command or RTCS to Send	Inhibit ID Status Bit to SET	Command or RTCS to Send
1-75	0-255	4.096 8.192 16.384 32.768 64.536	0-63	DISABLE, LATCH, CONTINUE	ENA/DIS	ENA/DIS	ENA/DIS	0-15	0-255	Complete Mnemonic or RTCS	0-255	Complete Mnemonic or RTCS
Example TBS by FOT	TBS by FOT	4.096	9	dis	dis	ena	ena	5		OOL1 RTCS#		OOL2 Send Commands

TMON Group #

Added at the time the TMON is generated.

TMON Group Inhibit ID

Number of the Inhibit Category the TMON is assigned to.

Execution Interval

Number of Seconds between TMON processing times.

Major Cycle Offset

First major cycle in which the TMON is executed within the

Master Cycle.

After Trigger
configured to:

After a TMON has performed its out of limits action, it can be

- 1.) DISABLE itself from any further processing
- 2.) LATCH itself by resetting the OOL Counter
- 3.) CONTINUOUS do not reset the OOL Counter

Inhibit Bit Setting

Enables the setting of an inhibit category bit on MAX OOL.

Command Output

Enables Command transmission on OOL MAX.

Log Output

Enables reporting to the SCC Activity Log on OOL MAX.

MAX OOL

Sets the number of OOL readings required to trigger the TMON.

Inhibit Id Status Bit to Set

Provides the ID of the inhibit category to be set on MAX

OOL.

Command or RTCS to Send

Gives the Command Mnemonic or RTCS ID to be sent on MAX

OOL.

1283 TMON Test Description Table

TMON Test Description					
Subgroup	Telemetry Mnemonic	Operator & Engineering Value	Operator & Raw Count	True Path	False Path
0	Mnemonic	>,<= value	>,<= Count	PNT, JSG, OOL, INL	PNT, JSG, OOL, INL
0	Mnemonic	>,<= value	>,<= Count	PNT, JSG, OOL, INL	PNT, JSG, OOL, INL
0	Mnemonic	>,<= value	>,<= Count	PNT, JSG, OOL, INL	PNT, JSG, OOL, INL
0	Mnemonic	>,<= value	>,<= Count	PNT, JSG, OOL, INL	PNT, JSG, OOL, INL
1	Mnemonic	>,<= value	>,<= Count	PNT, JSG, OOL, INL	PNT, JSG, OOL, INL
1	Mnemonic	>,<= value	>,<= Count	PNT, JSG, OOL, INL	PNT, JSG, OOL, INL
1	Mnemonic	>,<= value	>,<= Count	PNT, JSG, OOL, INL	PNT, JSG, OOL, INL
2	Mnemonic	>,<= value	>,<= Count	PNT, JSG, OOL, INL	PNT, JSG, OOL, INL
2	Mnemonic	>,<= value	>,<= Count	PNT, JSG, OOL, INL	PNT, JSG, OOL, INL
3	Mnemonic	>,<= value	>,<= Count	PNT, JSG, OOL, INL	PNT, JSG, OOL, INL
3	Mnemonic	>,<= value	>,<= Count	PNT, JSG, OOL, INL	PNT, JSG, OOL, INL
3	Mnemonic	>,<= value	>,<= Count	PNT, JSG, OOL, INL	PNT, JSG, OOL, INL
3	Mnemonic	>,<= value	>,<= Count	PNT, JSG, OOL, INL	PNT, JSG, OOL, INL

Subgroup There can be up to 4 subgroups in each TMON.
 Telemetry Mnemonic This needs to be the complete telemetry Mnemonic.
 Operator & Engineering Value The operators available are >,<= . The engineering value should be supplied here.
 Operator & Raw Count The operators available are >,<= . The raw count should be supplied here.

 True Path This is the Path option for the True Path.
 False Path This is the Path option for the False Path.

 Path Options:
 1. PNT Perform the next test in the subgroup.
 2. JSG Jump to the next subgroup.
 3. OOL Exit the group on an OOL condition.
 4. INL Exit the group on an INL condition.

13. Flight Software (FSW) Patch SOP

131 Originator

13.1.1 Name: John Teter Date: 8/1/97

13.1.2 Revised by: Date:

132 Purpose

The Purpose of this Standard Operating Procedure (SOP) is to document the procedure for patching the Flight Software (FSW). This SOP will describe the creation, testing, and uplink of a FSW Patch. It will also describe the responsibilities of the parties involved in patching the Flight Software, namely the Flight Software Systems Branch (FSSB) and the Flight Operations Team (FOT).

133 Background

Flight Software Patches are required periodically to maintain proper spacecraft performance. These patches are different from table loads that affect the operation of Flight Software (i.e. TMON, RTCS). These patches are performed on the FSW code itself, and therefore must be performed with thorough testing to ensure that any software bugs are isolated before the patch is uplinked. It is equally important that a dump image of the patch be stored on the ground for analysis.

134 Responsibilities

13.4.1 FOT Responsibilities

The FOT is responsible for providing dump images to the FSSB, in order to assist them in FSW maintenance. The FOT will provide the FSSB any data necessary for generating the patch and assist the FSSB in testing the FSW Patch. The FOT will convert the supplied Colorado System Test and Operations Language (CSTOL) to EOS Command Language (ECL). The FOT is responsible for the uplink of the FSW Patch under the supervision of FSW maintenance personnel.

13.4.2 FSSB Responsibilities

The FSSB personnel are responsible for creating the FSW Patch along with the Backout Patch, and testing them on the FTSB. FSW maintenance personnel will provide a CSTOL procedure for the uplink of the FSW Patch. FSW maintenance personnel are responsible for overseeing the uplink of the FSW Patch.

135 Prerequisites/Constraints

The following activities must be completed prior to uplinking a patch.

1. The FSW Patch must have been successfully tested using the FSTB.
2. The Uplink procedure must have been converted to ECL.
3. The FSW Patch must have been reviewed and approved by the FOT Operations CCB

136 Procedures

13.6.1 Creation of a FSW Patch

FSW Patches will be created by the FSSB Personnel responsible for FSW maintenance. FSW Patches may be created to enhance spacecraft performance, or to correct an anomalous condition. A backout Patch must also be created in order to correct any problems that may occur when the new patch is installed.

13.6.2 Transfer of the FSW Patch to the EOC

The FSW patch along with the backout patch, will be transferred from the FSTB to the EOC via an IST connection. The transfer is carried out using both the EOSDIS Backbone Network (EBnet), and the Center Network Environment (CNE). Likewise transfers of FSW Dumps can be sent from the EOC to the FSTB using the same path.

13.6.3 Generation of a FSW Load

Flight Software Loads will be generated using the FOS Binary Load Builder Tool. The binary file which was transferred from the FSTB will have to be ingested into the Binary Load Builder Tool before the load can be generated.

13.6.4 Testing of the FSW Load

After the FSW load has been generated it should be loaded into the Spacecraft Simulator (SSIM) and evaluated. The simulation should test the entire functionality of the patch. Likewise the backout patch should be tested on the SSIM also. It should be able to return the spacecraft software to its nominal configuration prior to the new patch being uplinked.

13.6.5 CCB Approval

After the load has been tested, it must be submitted to the FOT CCB for approval. Any errors or shortcomings of the patch should be corrected by this stage. The back out patch must have already been tested and must be ready for uplink if a problem should occur. Once the CCB approves the FSW Patch it can be uplinked to the spacecraft under FSSB supervision.

13.6.6 Uplink of the FSW Load

The FSW load must be uplinked with a FSSB representative present. The uplink procedure that is supplied by the FSSB FSW maintenance personnel should be followed to perform the FSW load. It must have been converted from CSTOL to ECL in order to be used on the FOS Ground System. If a problem occurs the Backout Patch should be loaded to restore the software to its previous state until the problem can be analyzed and corrected.

137 References

1. FOS Operations Tools Manual for the ECS Project 609-CD-005-002
2. FOS Integration and Testing Procedures
3. FOS Critical Design Review
4. FOT Configuration Management SOP
5. ICD between ECS and SDVF
6. EOS AM-1 MOR Review

138 Appendix

N/A

14. Decision Support System (DSS) Modification SOP

141 Originator

14.1.1 Name: Mark Heidenreich **Date:** 8/26/97

14.1.2 Revised by: **Date:**

142 Purpose

The purpose of this Standard Operating Procedure (SOP) is to document the process for creating and/or modifying a Decision Support System (DSS) rule or configuration. This SOP provides the user with a description of the DSS and its components. A description of the process to change the DSS configuration is included down to the code level. The Configuration Management (CM) aspect of this SOP is covered in an SOP dedicated strictly to CM issues.

143 Background

The DSS is a real-time automated assistant for the EOS Operations Center (EOC). The purpose of the DSS is to assist the on-line spacecraft operations personnel in evaluating the health & safety of the satellite. The DSS is an expert system running in the Talarian RTWorks rule based environment.

The DSS has access to spacecraft telemetry through the Flight Operations Segment (FOS) real-time telemetry server. The DSS takes the telemetry from FOS and checks the mnemonics specified in the rules. The RTWorks rules are just a series of “if then” checks which are used to identify a certain condition and take the corresponding action when that condition is met. A sample rule is provided in the appendix. (Figure 14.8-1.)

The DSS also consists of code which connect the RTWorks system to the FOS. This can be referred to as FOS supporting software. In addition to this code and the RTWorks software,

there are Perl and TCL scripts which execute in the DSS environment to perform different functions. (For examples of Perl and TCL scripts, see Appendix Figures 14.8-2 and 14.8-3) Any changes to these software items will require the approval of the Configuration Control Board (CCB).

The DSS is a rather complex system which is systematically easy to change, particularly the rule base. The changes can cause unforeseen problems in other portions of the system, so caution should be used when modifying existing code. When the idea for a change arises, the appropriate subsystem engineer should be made aware of the idea and how it would fit into their particular subsystem. If the subsystem engineer agrees that a worth while change can be made, then the code is developed with the assistance of the FOT DSS Engineer. When that step is complete, a test needs to be performed on the new software to verify that the appropriate functionality is actually achieved. How this test is performed will be dependent on the scope of the individual change and the details will be worked out accordingly. After testing is complete, the rule change is submitted to a Configuration Control Board (CCB) for review.

144 Responsibilities

The FOT will have an engineer whose responsibility it is to maintain the DSS. This person will be responsible for assisting spacecraft subsystem engineers in verifying the operational viability of rules which the subsystem engineers want to add, delete or change in the DSS. The subsystem engineers are responsible for understanding how the DSS is performing on their individual subsystem, but the DSS engineer is responsible for rule logic confirmation, script enhancements and DSS system architecture maintenance. The FOT CCB will be responsible for the overall CM of the DSS.

145 Prerequisites/Constraints

When a rule change is selected for testing, it must conform to the DSS system design constraints as established by the FOT DSS Engineer and the CCB.

1. The rule must conform to the FOT defined structure for DSS rules. This structure will be defined in DSS documentation delivered by the FOS.
2. The rule must conform to the constraints leveled by RTWorks, FOS and any other existing software in the DSS.
3. All relevant information with respect to the new rule and its incorporation into the DSS must be clearly identified and documented. This includes spacecraft specific modes/configurations, orbital events/anomalies and nominal performance expectations for the rule.

146 Procedure

14.6.1 Identify DSS Change Necessity

The person identifying the need for a DSS change should discuss the change with the appropriate subsystem engineer and the DSS Engineer. A change in this discussion could be to any one of the 4 areas (FOS supporting software, RTWorks code, TCL or Perl scripts) of DSS code. If the discussion leads to further pursuit of a change, then the process continues, else the idea ends here. The purpose for this step is to verify that the idea doesn't already exist or that the idea is feasible in the DSS environment.

14.6.2 Modify the System

The idea now must be transformed into a rule set or script for the DSS. A rule set should conform to the FOT defined standard rule architecture unless there is a good reason to vary. The rule set must comply with RTWorks syntax. The rule set is written and checked against the DSS system for syntax errors. A script will conform to any set standards for scripts as defined by the FOT. The scripts will also be checked against their respective systems for syntax errors. All of the appropriate telemetry and external information must be identified and incorporated in supporting documentation or comments for future reference.

14.6.3 Modified System Testing

A new rule set should be incorporated into the DSS system and then the appropriate test performed against it. There will be several different methods available for testing. The new rule base can be run against the Spacecraft Simulator (SSIM), the EOSDIS Test System (ETS) or against the spacecraft itself. The latter can be done either while on the ground (engineering test data) or on orbit. When the new rule base passes all predetermined tests with the expected results, this portion of the process is complete. The same series of test options is available to script testing. Each change will require a certain level of testing which is solely dependent on that change.

14.6.4 Submit CCR & Validation Form

The system modification must now be submitted to the CCB with a Configuration Change Request (CCR) form. The CCR must contain a description of the change along with a reason. The CCR should be submitted with the test results as proof of validation.

14.6.5 CCB Review

Once the CCB receives the CCR and accompanying data, they will make a final decision on whether the change will be incorporated into the DSS. If the change is accepted, the FOT DSS Engineer performs the final steps to incorporate the change. If the change is rejected, the process is ended here.

147 References

1. Perl : Learning Perl, Author: Larry Wall, Publisher: O'Reilly and Associates
Programming Perl, Author: Larry Wall, Publisher: O'Reilly and Associates
2. tcl/tk : Tcl and the Tk Toolkit, Author: John Ousterhout, Publisher: Addison-Wesley
3. RTWorks : RTie inference engine, Publisher: Talarian. (available only from Talarian)

148 Appendix

1. DSS Rule Example
2. Perl Script Example
3. Example TCL Script

14&1DSSRuleExample

```

/*                                     */
/* DSS ADAC all mode related rules   */
/*                                     */

/*****
/*                                     */
/*           ACE related check block           */
/*                                     */
*****/

/*                                     */
/* ADAC all mode ACE check                               */
/*                                     */
/* slots : dss_adac.ace.str                               */
/* states : Nominal/Suspect/UNKNOWN                               */
/*                                     */
/* Comments/Notes :                               */
/*                                     */
/* PDL :                               */
/*                                     */
NAME: "ADAC ACE nominal check rule";
COMMENT: "Determine if ACE is nominal";
AUTHOR: "Pete Chui";
IF dss_tq.tq_top.str = "nominal"
AND Known(dss_cfg.shm_ace.str)
AND (gnc.am1_gnc_sr_shdp?side?_seuerr.value = 0
AND gnc.am1_gnc_sr_shdp?side?_ramtest.value = 0
AND gnc.am1_gnc_sr_shdp?side?_ad_fail.value = 0
AND gnc.am1_gnc_sr_shdp?side?_majvote.value = 0)
AND dss_cfg.shm_ace.str = ?side?
THEN dss_adac.ace.str := "nominal";
dss_adac.ace.msg_type := "INFO";
dss_adac.ace.msg_text := ConCat("ACE ",Uppercase(dss_cfg.shm_ace.str)," nominal");
!

NAME: "ADAC ACE suspect check rule";
COMMENT: "Determine if ACE is suspect";
AUTHOR: "Pete Chui";
IF dss_tq.tq_top.str = "nominal"
AND Known(dss_cfg.shm_ace.str)
AND (gnc.am1_gnc_sr_shdp?side?_seuerr.value = 1
OR gnc.am1_gnc_sr_shdp?side?_ramtest.value = 1
OR gnc.am1_gnc_sr_shdp?side?_ad_fail.value = 1
OR gnc.am1_gnc_sr_shdp?side?_majvote.value = 1)
AND dss_cfg.shm_ace.str = ?side?
THEN dss_adac.ace.str := "suspect";
dss_adac.ace.msg_type := "ALERT";
dss_adac.ace.msg_text := ConCat("ACE ",Uppercase(dss_cfg.shm_ace.str)," suspect");
!

```

```
NAME: "ADAC ACE cleanup check rule";
COMMENT: "ACE check cleanup";
AUTHOR: "Pete Chui";
IF dss_tq.tq_top.str <> "nominal"
OR NOT KNOWN(dss_tq.tq_top.str)
/* OR dss_cfg.ctrl_mode.str <> "shm_on" */
THEN dss_adac.ace.str := UNKNOWN;
dss_adac.ace.msg_text := UNKNOWN;
dss_adac.ace.msg_type := UNKNOWN;
ObjClear("dss_adac","ace");
!
```

1482PerlScriptExample

```
#!/tools/bin/perl

# Setup
$mcs_path_env = "MCSHOME";    # define default mcs path
$log_path = "/logs/process/"; # define default log files path
$log_filename = "dss.MH";     # define default MH filename
$out_filename = "dss.log";    # define default output filename

# find environment variable MCSHOME value
while (($key,$value) = each(%ENV)) {
    if ($key =~ /\b$mcs_path_env\b/) {
        $mcs_basepath = $value;
    }
}

# build full paths
$full_log_file_path = $mcs_basepath . $log_path;
# open log file
$log_file = $full_log_file_path . $log_filename;
$out_file = $full_log_file_path . $out_filename;
open(LOFIL, ">$out_file") || die("could not open file ($out_file).");
open(LIFIL, "$log_file") || die("could not open ($log_file).");

# Put header in output file
print LOFIL "    DSS State Recognition Engine (SRE) Log  \n";
print LOFIL "    ----- \n";

# Removed header stuff
while (<LIFIL>) { # !EOF ??
    @hline = split(/\s+/, $_); # parse header line
    if (@hline[0] ne "Time") { # removed all header lines ??
    }
    else {
        last;
    }
}

# Remove time, frame, and executing lines
while (<LIFIL>) { # !EOF ??
    $rline = $_;
    @rline = split(/\s+/, $rline); # parse input line
# check for initial state messages
    if ((@rline[0] eq "received") && (@rline[1] eq "info:")
        && (@rline[4] eq "System") && (@rline[6] eq "initial")) {
        # output system line as is
        $new_time = @rline[3];
        if ($new_time ne $old_time) { # separate different times
            print LOFIL "\n \n";
        }
        $str_ptr = index($rline,":");
    }
}
```

```

    $nrline = substr($rline, $str_ptr+2);
    print LOFIL $nrline;
}
# check for PALS generated messages
elseif ((@rline[0] eq "received") && (@rline[1] eq "info:")
&& (@rline[2] eq "PAL:")) { # Output info line as is
    $new_time = @rline[3];
    if ($new_time ne $old_time) { # separate different times
        print LOFIL "\n\n";
    }
    $str_ptr = index($rline, ".");
    $nrline = substr($rline, $str_ptr+2);
    print LOFIL $nrline;
}
# check for Delta messages
elseif ((@rline[0] eq "received") && (@rline[1] eq "info:")
&& (@rline[4] eq "(Delta)")) { # output system line as is
    $new_time = @rline[3];
    if ($new_time ne $old_time) { # separate different times
        print LOFIL "\n\n";
    }
    $str_ptr = index($rline, ".");
    $nrline = substr($rline, $str_ptr+2);
    print LOFIL $nrline;
    $srline = <LIFIL>;
    $srllen = length($srline);
    if ($srllen > 1) { # do not print blank lines
        print LOFIL $srline;
    }
}
# check for standard messages
elseif ((@rline[0] eq "received") && (@rline[1] eq "info:")
&& (@rline[4] eq "System")) { # output system line as is
    $new_time = @rline[3];
    if ($new_time ne $old_time) { # separate different times
        print LOFIL "\n\n";
    }
    $str_ptr = index($rline, ".");
    $nrline = substr($rline, $str_ptr+2);
    print LOFIL $nrline;
    $srline = <LIFIL>;
    $srllen = length($srline);
    if ($srllen > 1) { # do not print blank lines
        print LOFIL $srline;
    }
}
# check for IPC messages
elseif ((@rline[0] eq "received") && (@rline[1] eq "info:")
&& (@rline[4] eq "Sharing")) { # Output info line as is
    $new_time = @rline[3];
    if ($new_time ne $old_time) { # separate different times
        print LOFIL "\n\n";
    }
}

```

```
    }  
    $str_ptr = index($rline,":");  
    $nrline = substr($rline, $str_ptr+2);  
    print LOFIL $nrline;  
  }  
# discard all other messages  
  else {          # discard all other lines for now  
  }  
  $old_time = $new_time;      # save for next line  
}  
  
close(LIFIL);  
close(LOFIL);
```

1483ExampleTCLScript

```

#!/data/mcs_2.01/apps/arch/sun4_solaris/bin/awish

# Setup
    AWdJoin status
    AWmShare

#
# Description : Displays current values of all telemetry points used
#               by defined systems
#
# Assumptions : The files with the telem list MUST BE IN
#               $MCSHOME/apps/pals for now
#
# Function : Reads a list of telemetry points from a file and
#            display results in MH.
#
proc tvals {args} {
    global env
    set mcshome $env(MCSHOME)
# file access setup
    cd $mcshome/apps/pals
    if {[llength $args] == 1} then {
        set fname [lindex $args 0]
    } elseif {[llength $args] == 0} {
        set fname "telem.list"
    } else {
        Log alert "tvals : incorrect number of arguments"
        Log alert " Valid number of arguments : Either 0 or 1"
        return 1
    }
    set fid [open $fname r]
# printout header
    Log info " Telemetry value dump : "
    while {[gets $fid newline] >= 0} {
        Log info " value for $newline is [eval AWdGet $newline] "
    }
    close $fid
}
#
# Description : Displays current values of all RTworks slots
#
# Function : Reads a list of RTworks slots
#
proc slots {args} {
    set all_known [lindex $args 0]
    if {$all_known == 0} then {
        AWdSet rt_slot "all"
    } else {
        AWdSet rt_slot "known"
    }
}

```

```

    }
}
#
# Description : Generates and then displays the current SRE log
#
# Assumptions : The SRE log is in $MCSHOME/logs/process for now
#
proc show_log {} {
    global env
    set mcshome $env(MCSHOME)
# file access setup
    cd $mcshome/apps/pals
    exec dss_log_parse
    exec textedit $mcshome/logs/process/dss.log &
}
#
# Description : Print the current SRE log
#
# Assumptions : The SRE log is in $MCSHOME/logs/process for now
#
# Note : Path of nenscript need to be updated !!!!!!!!!!!
#
proc print_log {} {
    global env
    set mcshome $env(MCSHOME)
# file access setup
    cd $mcshome/logs/process
    exec /data/jbrown/bin/nenscript -1r -G -N -t4 -c -Psandshark dss.log
}
#
# Description : Filter ADAC messages
#
# Assumptions : None
#
proc adacmsg {args} {
    global adacmsgv
    set set_reset [lindex $args 0]
    if {$adacmsgv == 1} then {
        AWdSet adac_msg "set"
    } else {
        AWdSet adac_msg "reset"
    }
}
#
# Description : Filter CFG messages
#
# Assumptions : None
#
proc cfgmsg {args} {
    global cfgmsgv
    if {$cfgmsgv == 1} then {
        AWdSet cfg_msg "set"
    }
}

```

```

        } else {
            AWdSet cfg_msg "reset"
        }
    }
#
# Description : Filter EPS messages
#
# Assumptions : None
#
proc epsmsg {args} {
    global epsmsgv
    if {$epsmsgv == 1} then {
        AWdSet eps_msg "set"
    } else {
        AWdSet eps_msg "reset"
    }
}
#
# Description : Filter Sim messages
#
# Assumptions : None
#
proc simmsg {args} {
    global simmsgv
    if {$simmsgv == 1} then {
        AWdSet sim_msg "set"
    } else {
        AWdSet sim_msg "reset"
    }
}
#
# Description : Filter TQ messages
#
# Assumptions : None
#
proc tqmsg {args} {
    global tqmsgv
    if {$tqmsgv == 1} then {
        AWdSet tq_msg "set"
    } else {
        AWdSet tq_msg "reset"
    }
}
#
# Description : Filter All messages
#
# Assumptions : None
#
proc allmsg {args} {
    global allmsgv
    if {$allmsgv == 1} then {
        AWdSet adac_msg "set"
    }
}

```

```

        AWdSet cfg_msg "set"
        AWdSet eps_msg "set"
        AWdSet sim_msg "set"
        AWdSet tq_msg "set"
    } else {
        AWdSet adac_msg "reset"
        AWdSet cfg_msg "reset"
        AWdSet eps_msg "reset"
        AWdSet sim_msg "reset"
        AWdSet tq_msg "reset"
    }
}

#
# GUI section
#

# Create DSS control window
toplevel .dssctrl
wm title .dssctrl "DSS Control"
wm iconname .dssctrl "DSS Ctrl"

frame .dssctrl.telem -relief groove -borderwidth 4
frame .dssctrl.state -relief groove -borderwidth 4
frame .dssctrl.log -relief groove -borderwidth 4
frame .dssctrl.msg -relief groove -borderwidth 4
frame .dssctrl.dbmsg -relief groove -borderwidth 4
frame .dssctrl.exit -relief groove -borderwidth 4

button .dssctrl.telem.all -text "Display Telemetry Values" -command "tvals"

button .dssctrl.state.all -text "Display all slots" -command "slots 1.0"

button .dssctrl.log.show -text "Display SRE log" -command "show_log"
button .dssctrl.log.print -text "Print SRE log" -command "print_log"

checkbutton .dssctrl.msg.msg1 -text "Display ADAC Msg" -variable adacmsgv
-command "adacmsg" -anchor w
checkbutton .dssctrl.msg.msg2 -text "Display Config Msg" -variable cfgmsgv
-command "cfgmsg" -anchor w
checkbutton .dssctrl.msg.msg3 -text "Display EPS Msg" -variable epsmsgv
-command "epsmsg" -anchor w
checkbutton .dssctrl.msg.msg4 -text "Display Sim Msg" -variable simmsgv
-command "simmsg" -anchor w
checkbutton .dssctrl.msg.msg5 -text "Display TQ Msg" -variable tqmsgv -command
"tqmsg" -anchor w
checkbutton .dssctrl.msg.msg6 -text "Display all Msg" -variable allmsgv
-command "allmsg" -anchor w

button .dssctrl.exit.exit -text "Exit" -command "exit"

```

```
pack .dssctrl.exit.exit -side left -expand 1  
pack .dssctrl.exit -side bottom -fill x
```

```
pack .dssctrl.msg.msg6 .dssctrl.msg.msg5 .dssctrl.msg.msg4 .dssctrl.msg.msg3  
.dssctrl.msg.msg2 .dssctrl.msg.msg1 -side bottom -expand 1 -fill x  
pack .dssctrl.msg -side bottom -fill x  
pack .dssctrl.log.show .dssctrl.log.print -side bottom -expand 1  
pack .dssctrl.log -side bottom -fill x  
pack .dssctrl.state.all -side left -expand 1  
pack .dssctrl.state -side bottom -fill x  
pack .dssctrl.telem.all -side left -expand 1  
pack .dssctrl.telem -side bottom -fill x
```

15. FOS Configuration (Files) Definitions SOP

151 Originator

15.1.1 Name: Franz M. Filicky

Date: 18 AUG 97

15.1.2 Revised by:

Date:

152 Purpose

The purpose of this FOS (Flight Operations Segment) Configuration (Files) Definitions Standard Operating Procedure (SOP) is to familiarize the Flight Operations Team (FOT) members with the FOS configuration files definition and location of the files.

153 Background

The FOS Configuration files were created by the developers of FOS software and are utilized for the configuration of the FOS software at system startup. The configuration files are located in directory: **/fosb/test/am1/config**

154 Responsibilities

It is the responsibility of the FOS CM's and/or the FOT CM to maintain these files. Currently the files are the responsibility of the FOS CM. The FOT CM will be assume responsibility once the software is delivered and accepted by the FOT. It will be the responsibility of the FOT Configuration Management (CM) Coordinator to implement any approved changes to the configuration files. Any changes to the configuration files will be approved by the Configuration Control Board (CCB) prior to the implementation of the changes.

155 Prerequisites/Constraints

TBD

156 Procedures

15.6.1 Definitions

The configuration files in directory: **/fosb/test/am1/config** are broken down by the subsystem that they pertain to. The first two letters of the configuration file name tell which subsystem the file pertains to.

- Fa - Analysis
- Fc - Command
- Fd - Database
- Fm - Command Management System

Fo - Files common to all FOS Subsystems
 Fr - String Manager
 Ft - Decom
 Fu - FOS User Interface

There are two files per type of configuration file; *Items.cnfg and *ConfigFile.cnfg or Configuration.cnfg. The *Items.cnfg is the description of what configuration is done by the *ConfigFile.cnfg or Configuration.cnfg file.
 (See Appendix for the listing of the configuration file with a brief description of each.)

157 References

1. FOS Operations Tools Manual for the ECS Project 609-CD-005-003
2. FOS Integration and Testing Procedures
3. FOS Critical Design Review
4. FOT Configuration Management SOP

158 Appendix

15.8.1 FOS Configuration Files

FaAcConfigItems.cnfg - Analysis Request Cruncher configuration file
 FaAlConfigFile.cnfg - Activity Log Monitor configuration file
 FaAlConfigItems.cnfg - Description of Activity Log Monitor configuration file
 FaOaConfigFile.cnfg - Common configuration file applicable to analysis functions
 FaOaConfigItems.cnfg - Description of Common configuration file applicable to analysis functions
 FaRtConfigFile.cnfg - Rtworks Dataserver Configuration file
 FaRtConfigItems.cnfg - Description of Rtworks Dataserver Configuration file
 FcCdFormatConfigItems.cnfg - Description of Command Format configuration file
 FcCdFormatConfiguration.cnfg - Command Format configuration file
 FdDbConfigItems.cnfg - Database configuration file
 FmCcAM1ConfigItems.cnfg - Description of CMS Constraint Checker configuration file for AM1
 FmCcAM1Configuration.cnfg - CMS Constraint Checker configuration file for AM1
 FmCcConfigItems.cnfg - Description of CMS Constraint Checker configuration file
 FmCcConfiguration.cnfg - CMS Constraint Checker configuration file
 FmGsConfigItems.cnfg - Description of Ground Scheduler configuration file
 FmGsConfiguration.cnfg - Ground Scheduler configuration file
 FmImAM1ConfigItems.cnfg - Definition of Memory Image configuration file for AM1
 FmImAM1Configuration.cnfg - Memory Image configuration file for AM1
 FmImConfigItems.cnfg - Definition of Memory Image configuration file
 FmImConfiguration.cnfg - Memory Image configuration file
 FmImDumpInitConfig.cnfg - Dump Init configuration file
 FmIpConfigItems.cnfg - Definition of Proxy (Communications between CMS processes) configuration file
 FmIpConfiguration.cnfg - Proxy (Communications between CMS processes) configuration file
 FmLdConfigItems.cnfg - Definition of Load Catalog configuration file
 FmLdConfiguration.cnfg - Load Catalog configuration file
 FmLdLoadInitConfig.cnfg - Load Initiate Command configuration file

FmMmConfigItems.cnfg - Definition of Spacecraft Model configuration file
FmMmConfiguration.cnfg - Spacecraft Model configuration file
FmOmConfigItems.cnfg - Definition of Common CMS configuration file
FmOmConfiguration.cnfg - Common CMS configuration file
FmScConfigItems.cnfg - Definition of Schedule Controller configuration file
FmScConfiguration.cnfg - Schedule Controller configuration file
FmUnConfigItems.cnfg -

NOTE: The following “Fo” files are Common Configuration files used by the FOS subsystems

FoCfConfigItems.cnfg - Definition of
FoCfConfiguration.cnfg -
FoCfDBConfigItems.cnfg - Definition of Database configuration file
FoCfDBConfiguration.cnfg - Database configuration file
FoCfRcmConfigItems.cnfg - Definition of
FoCfRcmConfiguration.cnfg -
FoFtConfigData.cnfg -
FoFtConfigItems.cnfg - Definition of
FoRfConfigItems.cnfg -
FoSwConfigData.cnfg - Port and IP address configuration file
FoSwConfigItems.cnfg - Definition of Ports and IP addresses configuration file
FoTvClientConfigItems.cnfg - Tivoli gateway client configuration file
FoTvConfigItems.cnfg - Tivoli gateway configuration file
FoTvServerConfigItems.cnfg - Tivoli gateway server configuration file
FrCfRmsConfigFile.cnfg -
FtDcConfigItems.cnfg -
FtMdConfigItems.cnfg -
FtScConfigItems.cnfg -
FuDm_Data.cnfg -
FuDm_Setup.cnfg -
FuLbTableLoadInit.cnfg -
FuRIRTSLoadInit.cnfg -
MISR_tgt_sites_input -
XMITSTATE:AM1100.stf
XMITSTATE:AM1101.stf

NOTE: Majority of files below are used for FOS system checkout. More information will be provided later as this SOP is “Work Still in Progress”

access.db -
access_out.db -
acqFailure.dat -
act.db -
act_cd.db -
act_out.db -
algParams.db -
bap.db -
bap_out.db -
bufferOrder.db -
checkinterval
cmdproc.db
cmpconfig.lis

command.db
config.dat
configcodes.db
configfile.rms
constraint.db
constraint1.db
constraint_cd.db
constraint_out.db
data.db - Terminal names
deconfigfile.rms - DEC equipment configuration files
defaultPkts.dat
delete_main_tgt_tbl.sql
delete_temp_tgt_tbl.sql
ephemeris
etlConfig.db
event.db
eventType.db
eventType_out.db
event_out.db
footvecs
gchw.db - Ground control hardware names
gcuser.db - Ground control user names
gimbal.db
gimbal_out.db
help/
help.db - FOS help files using Netscape
hgaEvent.db
hw.db - terminal names
instr_tgt_file.fmt
lq.db
maPkts.dat
mode.db
mode_out.db
msgType.db
msgType_out.db
oneCATargetInputFile
oneExp
oneExpFoot
oneFPTargetInputFile
orb.das.los.gnd.bmd.db
outputSubSatPoints
plan.db
plan_out.db
readme.eg
respec_params.db
role.db - User role configuration file
rs.db
rsAssoc.db
rsAssoc_out.db
rs_out.db
rtnChTimeDly.dat
saMaPkts.dat

saPkts.dat
schRs.db
schRsAssoc.db
schRsAssoc_out.db
schRs_out.db
simPkts.dat
srs.db
state.db
state_out.db
stateevent.db
statesun.db
statetdrs.db
sunconfigfile.rms - SUN equipment configuration file
supcnfg.dir
supiden.db
target_table_transfer.sql
tdrsEvent.db
testParams.db
tgt_file_to_db.script
tgt_tbl_trans_fos_dba.sql
tgt_to_fos_dba.script
timeXfer.dat
user.db - User name database file

16. Room Builder SOP

161 Originator

16.1.1 Name: Alan D. Lampe

Date: 8/11/9797

16.1.2 Revised by:

Date:

162 Purpose

The purpose of this Room Builder Standard Operating Procedure (SOP) is to document the Flight Operations Team manner by which rooms are created, utilized and maintained in accordance with established Configuration Management (CM) procedures.

163 Background

The Flight Operations Team/Instrument Operations Team (FOT/IOT) can use the Room Builder to create and modify room definitions. Each room may contain one or more windows or tools, and such definable parameters as window size, position on screen, and window state (default or tiled) can be established and set under a given room name. The room may be saved either temporarily (for current session only) or permanently for future sessions.

164 Responsibilities

It is the responsibility of each FOT/IOT member to be familiar with the functions of the Room Builder. The proper use and application of the Room Builder will not only streamline the tasks of each FOT member, but also greatly increase efficiency in both a real-time and post-pass environment.

165 Prerequisites/Constraints

A display page must be approved by the Configuration Control Board (CCB) and properly defined in the Project Data Base (PDB) before it can be used in a room (see Display Builder SOP).

166 Procedures

16.6.1 Room Builder Elements

Users can define rooms by utilizing the FOS supplied Room Builder tool. The Control Window is used in display page and tool selection. Each room contains a Control window which is used to select the desired display pages and tools that are to be contained within the room.

16.6.2 Room Definition

Select TlmWins from the Control Window to access the menu of available display pages and select any pages that are to be contained within the room definition. Move and resize these windows into your desired room configuration. If desired, the room may contain pages that are in an iconified state.

16.6.3 Saving Room Definitions

Once the newly created or redefined room is completed, then that room can be given a name and that name can in turn be saved into the data base. Click Tools from the Control window and then select Room Builder from the menu. Enter the desired name in the Name text box of the Room Builder window. To save the room for future sessions, click Permanent. If the newly defined room does not need to be saved for future sessions, then click Temporary and the room will be saved for the current session only. To close the Room Builder tool click OK. The newly created room will now be available under the Rooms button on the Control Window.

16.6.4 Modifying an Existing Room

Enter the desired room and make your desired modifications. Click Tools on the Control Window and select Room Builder from the list of tools. Enter the room's name in the Name text box of the Room Builder. Save the newly modified room as specified in 16.6.3.

16.6.5 Deleting an Existing Room

Enter the desired room and then click Tools on the Control Window and select Room Builder from the list of tools. Click the Delete button and the selected room will be removed from the data base.

16.6.6 CM Process

A CCR (Configuration Change Request) is to be submitted along with a Configured Item (CI) listing and Validation Form for FOT CCB approval with each new room that is created or modified. The CCB will then review the CI and the Ops Manager, Spacecraft Manager, NASA Flight Ops Director (FOD) and CCB Chairperson will then sign the CCR indicating that it has been accepted.

167 References

1. FOS Operations Tools Manual for the ECS Project 609-CD-005-003
2. FOS Integration and Testing Procedures 322-CD-101-002
3. FOS Critical Design Review

168 Appendix

N/A